Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

1978-12

Simulation of tactical alternative responses (STAR)

Wallace, William Scott; Hagewood, Eugene Gordon

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/41682

SIMULATION OF TACTICAL
ALTERNATIVE RESPONSES
(STAR)

William Scott Wallace

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

SIMULATION OF TACTICAL
ALTERNATIVE RESPONSES
(STAR)

by

William Scott Wallace

and

Eugene Gordon Hagewood

December 1978

Thesis Advisors: S. A. Perry
E. P. Kelleher

Approved for public release; distribution unlimited

T186188

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Simulation of Tactical Alternative Responses (STAR)		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1978
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) William Scott Wallace Eugene Gordon Hagewood		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California		12. REPORT DATE December 1978
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

    This thesis presents a stochastic simulation model of
ground combat. The tactics represented in the model are
explained in detail, and a brief explanation of the SIMSCRIPT
programming language used in the model is presented. The
model is explained in detail, and the computer routines
which make up the model are included. The input requirements

DD _{1 JAN 73} FORM 1473　　EDITION OF 1 NOV 55 IS OBSOLETE
S/N 0102-014-6601 |

(continuation of abstract)

for execution of the model are explained in detail so that
this thesis might become the initial user's manual for
future applications of the model. A typical simulated
battle is presented with detailed explanation of the output
to enable the reader to better appreciate the potential
applications of the model. The current status of the
model referenced in the thesis is prior to the version
which will be used for production runs.

Simulation of Tactical
Alternative Responses
(STAR)

/   by

William Scott Wallace
Captain, United States Army
B.S., United States Military Academy, 1969

and

Eugene Gordon Hagewood
Captain, United States Army
B.S., Georgia Institute of Technology, 1969

ABSTRACT

This thesis presents a stochastic simulation model
of ground combat. The tactics represented in this model
are explained in detail, and a brief explanation of the
SIMSCRIPT programming language used in the model is
presented. The model is explained in detail, and the
computer routines which make up the model are included.
The input requirements for execution of the model are
explained in detail so that this thesis might become the
initial user's manual for future applications of the model.
A typical simulated battle is presented with detailed ex-
planation of the output to enable the reader to better
appreciate the potential applications of the model. The
current status of the model referenced in this thesis is
prior to the version which will be used for production
runs.

TABLE OF CONTENTS

6

LIST OF TABLES

LIST OF FIGURES

I.   INTRODUCTION

During the past several decades, the nature of battle has changed, not abruptly, but nonetheless significantly. The fact that the modern battlefield will be dominated by highly lethal weapons is well known. Technologically advanced supporting systems (combat support, combat service support, communications, etc.) are being developed and sent to the field. The high cost of these systems, either weapons or supporting systems, make it imperative that once the procurement process is complete, the best possible systems are introduced into the inventory. Such systems must meet the most stringent requirements for effectiveness, trainability, and maintainability. In recent years one of the most useful tools in evaluating proposed and existing systems has been the combined arms simulation model. With these facts in mind, the Simulation of Tactical Alternative Responses (STAR) combined arms model discussed in this document has been developed.

One of the primary goals of STAR is to achieve an acceptable level of resolution while assuring that the model inputs, interactions, and outputs are understood by the military decision maker. It is the intent of this thesis to outline the structure of STAR in sufficient detail to allow analysts and decision makers to fully understand the execution logic of the model, and thus understand the results of the model and how these results were derived.

The structure of STAR is truly hierarchical in that it is not confined to any specific unit size configuration. The SIMSCRIPT programming language has been chosen as the primary language for STAR. The reasons for selection of SIMSCRIPT and a brief discussion of those SIMSCRIPT programming features which are utilized in the model are discussed in Chapter II.

The BLUE and RED tactics modeled in the current version of STAR are discussed in chapter III. This chapter describes the assumptions made in regard to the tactics employed by individual weapon systems and unit organizations. The chapter also states, in general terms, how these tactics have been modeled.

Chapter IV is an in-depth discussion of the SIMSCRIPT events and routines which make up the model. Included in chapter IV is a discussion of each individual routine, an explanation of the variables used in the routine, a listing of the computer code for the routine and a line by line explanation of what the code does. Also included in chapter IV is a discussion of the PREAMBLE and MAIN programs of the model which prepare the model for execution.

Chapter V is a detailed discussion of the input requirements of the model. Included are specifications of the SIMSCRIPT inputs required and the job control language required to run the model. Proper formatting of the input and an explanation of the meaning of each block of input is also included.

Chapter VI is a simulated battle run using the current version of the STAR model. The progress of the battle is traced over time to show the reader the format of the model output and allow the reader to see what type of output could be expected from the model.

The current version of STAR utilizes a parametric terrain model developed by Needels [Ref. 8] while a student at the Naval Postgraduate School, and since enhanced by Needels and Professor James K. Hartman of the Naval Postgraduate School. The parametric terrain model provides a truly continuous terrain as opposed to the digitized terrain used in some existing models.

The model addressed in this thesis is, in a sense, an interim product. Work is continuing at the Naval Postgraduate School on a dynamic artillery module to be included in the model by mid-December 1978. Dynamic play of BLUE and RED air defense and close air support, as well as a detailed communications module, will be available by mid-April 1979. Enhancement of the existing model will continue through mid-1979 with the ultimate goal of providing a BLUE Brigade versus RED Division simulation which employs all those modules mentioned above plus the dynamic play of resupply at the unit and individual vehicle level. These enhancements are described in chapter VII.

Appendix A is a flowchart which represents the event flow of a single tank as the tank cycles through the simulation. Appendix B is a glossary of terms used in the thesis.

Throughout the thesis, repeated references are made to the "current version of STAR". The purpose of this thesis is to describe the logic flow of the model within the context of the current model status prior to incorporation of many detailed features required for production runs. Therefore, the reader must not consider this thesis as a documentation of the production version of the model. Such a document will be produced as a Naval Postgraduate School Technical Report in early 1979.

In order to exercise the logic of the model, many assumptions were made concerning tactics employed by both the attacking and defending forces. These assumptions were made based on the experience of the authors, both combat and non-combat, and were in general validated by an appropriate agency of the United States Army Training and Doctrine Command. Tactical decision making, combat intelligence, command and control, and tactical communications networks are not specifically modeled, but are addressed in the discussion of tactical assumptions. It should be obvious that this model is not a full representation of actual combat. Modeling simplifications have been made in several areas. However, none of these simplifications should be considered critical to the validity of the model.

II. THE USE OF SIMSCRIPT IN STAR

General

The SIMSCRIPT language, level II.5, was selected for
STAR because the language is designed for discrete event
simulation applications. Because of the many imbedded
features in SIMSCRIPT, the programmer is given opportunities
in the construction of event flow which are not available
in FORTRAN or other computation languages. SIMSCRIPT is
also similar to English in regard to the construction of
commands and the implementation of logical tests. This
feature, coupled with an essentially free form of syntax,
allows the code to be written in a very readable format.
This will become apparent when the coding for the events
and routines in STAR is introduced.

Also important to the programmer is the similarity of
SIMSCRIPT to FORTRAN with regard to computational syntax.
A programmer with a moderate amount of experience in
FORTRAN should have very few difficulties understanding and
writing computational routines in SIMSCRIPT.

Finally, since a wide variety of computational routines
exist in FORTRAN (terrain models, line of sight routines,
smoke models, lethality packages, etc.), it is important
that any simulation provide some reasonable interface with
these existing modules. SIMSCRIPT provides this capability

13

by allowing FORTRAN routines to be called as subroutines by SIMSCRIPT events and routines.

This last feature made SIMSCRIPT an even more appealing language, since existing FORTRAN routines can be used with only minor modifications. The difficult line of sight calculations, for example, are accomplished in FORTRAN because the terrain model used by STAR was originally written in FORTRAN. Other routines which were more closely tied to the structure of SIMSCRIPT were written in that language. The routine that updates the list of detected targets for each TANK, for example, is written in SIMSCRIPT to take advantage of the dynamic dimensioning capability of the language and the pointer value link listing techniques available. For large target arrays, these SIMSCRIPT language features are extremely efficient in reducing core storage requirements and execution time.

Main Structural Components of SIMSCRIPT

The primary imbedded simulation facilities available to STAR are structural components known as entities, attributes, sets, and events. These components are used in conjunction with an internal timing mechanism which is referred to as the system timer throughout this thesis. These facilities greatly simplify the process of writing a simulation program and debugging code. The process is further simplified by a compiler which provides error messages and traceback routines similar to the WATFOR and WATFIV compilers in FORTRAN.

14

The Structure of STAR

The structure of STAR begins with the concept of an entity. An entity is simply a representation or model of an item to be simulated. In other simulations, entities might represent people waiting in line for tickets, customers in a bank, or automobiles waiting to be serviced at a gas station. In STAR the basic entities are weapon systems representing tanks, anti-tank guided missiles, artillery pieces, and are referred to generally as TANKs. Throughout this thesis the following terms are synonomous with the notion of an entity: TANK, POINTER, FOE, element, weapon system, firer, target, and observer. Those terms which are capitalized are explicitly used as variables in the program. An entity may be brought into existence by using a simple phrase which includes the variable name of the entity. For example, the phrase CREATE A TANK reserves a place in core storage for the entity which the programmer has chosen to call TANK. Associated with the word TANK is a pointer value which defines the location in core storage where this particular TANK and information pertinent to it are stored.

It is desirable to associate certain characteristics with entities after they have been created. These characteristics are referred to as attributes and are affixed to an entity by the programmer or, as will be seen in the discussion of sets, may be placed on the entity by SIMSCRIPT bookkeeping procedures (referred to as the system). As an example of user affixed attributes, the programmer may wish

to assign a current X coordinate to a TANK. This is accomplished by the phrase: LET X.CURRENT(TANK) = 3150. Attributes may be changed during program execution to reflect changes in characteristics. The system changes system defined attributes as required.

In addition to associating characteristics with particular entities, it is often useful to group entities with similar characteristics. STAR uses sets to keep track of all TANKS on the battlefield for organizational and fire control purposes which enhances the programmer's ability to model unit tactics at any organizational level. An entity may belong to any number of sets and may also own sets. The following phrases illustrate set membership: FILE TANK IN TANKS, FILE TANK IN COMP.UNIT(5). In this example the TANK is first made a member of a set called TANKS. The system automatically assigns to the TANK a membership attribute, M.TANKS, whose value is the integer 1. The second FILE command makes the TANK a member of company number 5. Again, a membership attribute (M.COMP.UNIT) is assigned a value of 1. Ownership of sets is accomplished by a phrase of the form: EVERY COMPANY.COMMANDER OWNS A COMP.UNIT. In addition, each entity filed in a set has attributes which indicate who the first and last members of the set are and how many members belong to the set. BLUE force elements are filed in platoon sets (PLT.UNIT), company sets (COMP.UNIT), and a set of operational tanks (BLUE.ALIVE). Although entities may be filed by ranking attributes, STAR uses a simple first-in first-out procedure. RED force elements

16

are assigned similarly except that their set of operational TANKs is called RED.ALIVE. All TANKs (XM1s, IFVs, ITVs, DIVAD guns, DRAGONs, T72s, BMPs, and ZSUs) are assigned to a set called TANKS which is used for final output purposes.

Each entity in STAR is modeled to reflect a flow of activities over time. In particular, each entity initiates, or undergoes, search, detection, target selection, firing, impact, or location update. These six events are scheduled dynamically based on the current tactical situation or by way of an appropriate probability distribution which models the period of time between events. When an event is scheduled for an entity, the SIMSCRIPT timer makes a record of the time that the event is scheduled to occur (in terms of overall simulated time) and the entity for which the event has been scheduled. Other characteristics of the event may be recorded in a manner similar to the assignment of attributes. At the appropriate simulated time, the event is executed unless cancelled by some logic provided by the programmer. For example, the scheduling of a fire event by one TANK on another could be accomplished by the phrase: SCHEDULE A FIRE(A,ID) IN X SECONDS. The variable A represents the pointer value of the firing TANK, while ID represents the pointer value of the target TANK. This event would be filed by the system in an event set which contained, among other things, the time that the event is to take place, the entities involved in the event and the event location with respect to other scheduled events. When X seconds had elapsed from the current simulated time, the

event would be executed and the consequences of the logic written in the event routine would occur. Event routines may in turn generate other events. The FIRE event in STAR, for example, causes the scheduling of an IMPACT event. IMPACT may then cause the scheduling of DETECT and TARGET.SELECT events.

The information provided here should prepare the reader for the detailed discussion of the event and routine coding which is given in chapter IV.

III.   TACTICS MODELED IN STAR

The purpose of this chapter is to describe the tactical
conventions modeled in STAR. Due to the unquantifiable
nature of some of the tactics described herein, especially
in the area of RED tactics, the tactics modeled in STAR
represent, in some cases, a "best guess" as to how to
represent the tactics of a particular system or unit organ-
ization. The actual computer code which models these
tactical considerations in the simulation is outlined in
detail in chapter IV.

In order to understand the tactics represented in the
model, it is first necessary to understand how the weapon
systems in the simulation are organized and to have some
knowledge of the attributes of each of the weapons systems.
As mentioned in chapter II, each weapon system simulated is
a member of the set TANKS, which is a list of every element
in the simulation. Each BLUE weapon system that is currently
active in the simulation is a member of the set BLUE.ALIVE,
likewise each active RED system is a member of a company set
and a platoon set which represent the unit to which he
belongs. Every weapon system in the simulation has attri-
butes which indicate to that element which weapon system
in the simulation represents his company commander, platoon
leader, and section leader. Additional attributes indicate
to the weapon system what type of ammunition he has available

19

and the number of rounds remaining. The weapon types and
ammunition types available to each weapon system, and the
numeric code used to represent these weapons and ammunition
types are presented in table I. With this general under-
standing of the set and attribute structure of the model we
proceed with the tactical discussion.

Search and Detection

Search for targets takes place iteratively throughout
the simulation at intervals specified by a time increment
variable, DELTA.T, provided by the user. The essence of the
search procedure is that every DELTA.T seconds of the simu-
lation every weapon system will check every opposing weapon
system that is still "alive" in the simulation. If the
observer has line of sight to the target, the time it would
take for the observer to detect the target is determined and
at that future time the detected target is placed in a list
of currently detected targets unique to that particular
observer. No detections of greater than DELTA.T seconds
are scheduled. This prevents detections from overlapping
into the next search event. Detection can take place
external to this normal detection process in one of two ways.
First, a target that is fired on is allowed to detect the
firer without going through the normal detection process.
Second, those weapon systems in close proximity to a
normally detected target are considered detected as well
and are placed in the observer's target list.

20

TABLE I

WEAPON SYSTEM AND AMMUNITION CODES

WPN. TYPE ATTRIBUTE CODE	WEAPON SYSTEM NAME	AMMUNITION CODE AND AMMUNITION ATTRIBUTE NAME			
		1 AP.TOW	2 HE.DRAG	3 AW1.OR.MSL3	4 AW2.OR.ADM
1*	XM1	APDS	HEAT	.50 Cal MG	---
2*	XM1	APDS	HEAT	.50 Cal MG	---
3	IFV	TOW	---	Bushmaster	---
4	ITV	TOW	---	---	---
5	DIVAD	---	---	35/40mm gun	---
6	DRAGON	---	DRAGON	---	---
7	T72	APDS	HEAT	12.7mm MG	---
8	BMP	ATGM	---	73mm gun	---
9	ZSU	---	---	37mm gun	---

Ammunition codes not utilized by a particular weapon system are indicated by 3 hyphens (---).

*Two XM1 weapon type codes have been provided to allow for both the XM1 with 105mm main gun and the XM1 with 120mm main gun.

Target Selection

The use of danger state tables in target selection and the computer code which simulates the selection of targets are explained in detail in later chapters. A brief explanation of the target selection process is given in this section. The distance from firer to target is currently divided into three range bands for each firing weapon system. These bands are: less than 1000 meters from the system, 1000-2000 meters from the system, and greater than 2000 meters from the system. The observer searches his list of detected targets and selects the highest priority target within the closest range band. Within range bands, targets of equal priority are selected based on closest range, which is the normal target selection process for a weapon system. This normal selection process may be overridden by special tactical considerations described in subsequent paragraphs. Current development of a more comprehensive target selection model is described in chapter VII.

A. BLUE WEAPON SYSTEMS

Terrain Utilization

It is assumed that the BLUE weapon systems in defense will take maximum advantage of the available terrain. BLUE systems will assume a turret defilade position while acquiring targets, move to a firing defilade position during firing, and, after firing a specified number of rounds, move to a full defilade position which protects the system from direct fire while loading ammunition, redistributing

ammunition, accomplishing emergency repairs or moving to an alternate firing position. The number of rounds fired which trigger a movement to a full defilade is dependent on the type weapon system. The length of time spent in full defilade is dependent on the task being accomplished and the user specified times for the task.

Individual Firing of XM1 or DIVAD gun in Ground Support Role

The XM1 is allowed to fire two or three rounds from its firing defilade position. The determination of whether two or three rounds are fired is dependent on the sensing of the rounds and is best illustrated by table II. HIT refers to a sensed catastrophic kill and MISS refers to any other result.

The same logic as illustrated in table II applies to the DIVAD gun with 35 (or 40)mm gun. Note that each firing of the DIVAD gun represents a burst of 10 rounds.

```
                    ┌──────────────┐
                    │ Fire at      │
            HIT     │ selected     │    MISS
         ┌──────────│ target       │──────────┐
         │          └──────────────┘          │
         │                                     │
   ┌───────────┐                         ┌──────────────┐
   │ Select    │                   HIT   │ Refire       │   MISS
   │ new       │                 ┌───────│ same         │────────┐
   │ target    │                 │       │ target       │        │
   └───────────┘                 │       └──────────────┘        │
         │                       │                               │
   ┌───────────┐           ┌──────────┐                   ┌──────────┐
HIT│ Fire at   │MISS       │ Move  to │                   │ Move  to │
 ┌─│ new       │──┐        │   full   │                   │   full   │
 │ │ target    │  │        │ defilade │                   │ defilade │
 │ └───────────┘  │        └──────────┘                   └──────────┘
 │                │
┌──────────┐  ┌───────────┐
│ Move  to │  │ Refire    │   MISS
│   full   │  │ same      │────────┐
│ defilade │  │ target    │        │
└──────────┘  └───────────┘        │
                  │                │
          HIT     │                │
          ┌───────┘                │
    ┌──────────┐           ┌──────────┐
    │ Move  to │           │ Move  to │
    │   full   │           │   full   │
    │ defilade │           │ defilade │
    └──────────┘           └──────────┘
```
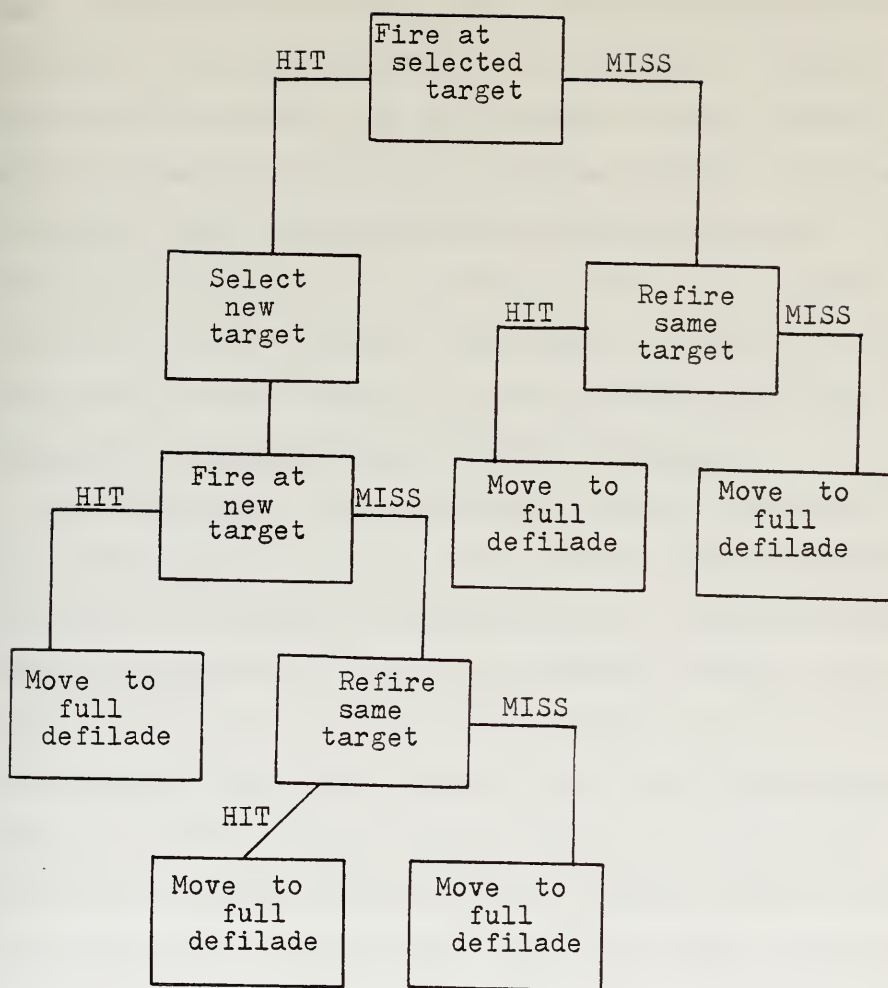
Table II   Round Sensing

24

Individual Firing of IFV, ITV, or DRAGON

The ITV, IFV, and DRAGON weapon systems go through the same sequence of turret defilade, firing defilade and full defilade as the XM1 and DIVAD. In the case of weapon systems which have no turret, the term turret defilade refers to the weapon system in a position of minimum exposure to direct fire while still allowing visual target acquisition. In the case of the ITV, IFV and DRAGON, the weapon system is allowed to fire two rounds regardless of sensing of round effect and is then moved to a full defilade position.

Selection of Targets While in Firing Defilade

The STAR model differentiates between a ballistic round and a wire guided round in terms of the firing system's selection of subsequent targets while in firing defilade. The logic employed is such that a system firing a wire guided round will always go through the process of selecting a new target (which may result in the same target being re-engaged), while a system firing a ballistic projectile will continue to engage the same target (without going through the target selection process) until the target is sensed as a catastrophic kill (at which time a new target selection takes place). In all cases, a movement to full defilade causes the weapon system's list of detected targets to be set to zero. The subsequent return to turret defilade and then to firing defilade causes the normal detection and selection process to take place.

Impact of ATGM (Anti-Tank Guided Missile) Rounds

The STAR model takes into account the eventuality of an ATGM round being fired and, before impact of the round, the firing system being destroyed. For both BLUE and RED missile firing systems which are killed with rounds in flight, the round is allowed to impact, but is always considered unguided at impact and assessed as a complete miss of the target.

Tank Platoon Fire Control

BLUE tank platoon fire control is modeled under the assumption that platoon positions are well planned and pre-pared, sectors of fire assigned to individual tanks and strict fire control and fire discipline observed. Platoon fire control is modeled such that no two tanks in the same platoon are allowed to engage the same target, with one possible exception. If a tank searches its entire target list and finds that all of his detected targets are being engaged by members of his platoon, he will engage the highest priority target in his list regardless of whether or not that target is currently engaged by another platoon member.

Mechanized Infantry Platoon Fire Control

BLUE mechanized infantry platoon fire control is modeled in a fashion similar to fire control for the tank platoon. No two IFVs within the same platoon are allowed to engage the same target simultaneously. The on board load of TOW rounds is assumed to be such that these rounds will be strictly controlled; thus, if all targets in a

selecting IFV's target list are engaged, that IFV will not
fire until such time as he detects a target not engaged by
another platoon member. The DRAGON teams in a mechanized
infantry platoon are assumed to be employed independently
and without advantage of communications with the platoon
headquarters. Thus, DRAGON teams are allowed to detect,
select, and fire on targets independent of the actions of
other platoon members.

Selection Fire Control for ITVs

The ITV sections attached to the company team are
assumed to operate under the same strict controls in regard
to TOW ammunition as the IFVs in the mechanized infantry
platoon. To model this, the ITVs in the company team are
all filed in the same platoon set and fire control is
modeled identically to that for the IFVs in the mechanized
infantry platoon.

BLUE Company Fire Control

Fire control at the BLUE company team level is not as
strictly modeled as are the fire control measures in the
platoons. Company team fire control and distribution is
accomplished by the careful selection of direction of
movement (DIR.OF.MVMT) and primary direction of search
(PRI.DIR) attributes for company team elements. These
attributes, when properly designated, allow for overlapping
and mutually supporting fields of fire for company team
elements. No other allowance is made for elements to co-
ordinate fires with weapon systems other than those in their
assigned platoon.

B. RED WEAPON SYSTEMS

Terrain Utilization

Red weapon systems in the offense are assumed to move
at a maximum speed allowed by the terrain and the enemy
situation, while maintaining formation integrity as much as
possible. Thus, the utilization of terrain by RED forces
is primarily through the selection of the best attack routes
to their assigned objectives. RED forces are assumed to
fire on the move in the attack and are thus not afforded
the option of seeking defilade while firing. The only ex-
ception is the case of the BMP while firing its ATGM. When
firing a wire guided missile, BMPs stop to fire and seek the
best cover and concealment along their movement route,
while maintaining formation integrity to the extent
possible.

RED Tank Platoon Fire Control

The RED tanks in the simulation are assumed to mass
fires on targets designated by the platoon leader. The RED
tanks fire on the move and fire on the target designated by
their platoon leader if the platoon leader is "alive", the
platoon member has line of sight with the platoon leader and
the platoon member has the proper type of ammunition avail-
able to engage the target. The failure to meet any of these
criteria causes a RED tank to go through the selection and
firing process independent of the leader's target. Upon
impact of a round, the RED tank goes through the target
selection process regardless of the sensing of the round.

28

Thus, RED tanks are not given the opportunity to sense the impact of a round and improve upon the hit probability of subsequent rounds fired at the same target. ZSUs in the ground support role use the same logic as RED tanks firing individually.

BMP Fire Control

Doctrinally, the overriding consideration of RED forces in the offense is the momentum of the attack. The primary role of the BMP is to serve as an infantry carrier. As a result, the BMP does not fire in the attack as long as the tanks of the unit to which it is attached provide protection for it. To model this tactic, the STAR model causes the BMP to select a target and check to see if any other weapon system in its company (to include other BMPs) is engaging that target; if so, the BMP does not fire at that particular target and selects another target from its list of currently detected targets. If every target in a BMP's list of detected targets is being engaged by other company members, the BMP continues to move without firing. The BMP stops to fire its ATGM but fires it 73mm gun on the move.

RED Company Fire Control

Fire control and distribution at the company level for RED forces is accomplished in the same manner as BLUE company level fire control.

IV. DETAILED EXPLANATION OF STAR

In this chapter each event and routine in STAR is explained in detail. With the exception of the PREAMBLE, the format used will include a description of the event or routine, including its purpose; a listing and definition of variables which are local to the event or routine; and a listing of the actual coding used in the current version of STAR, followed by a brief explanation of the code.

The PREAMBLE is somewhat different in structure and the primary area of interest is in the definition of the variables, arrays, entities, attributes, and sets which are global to all STAR events and routines.

A. PREAMBLE
General

The PREAMBLE is used to define global variables and arrays; declare entities, attributes, and sets; define events and FORTRAN routines; and prepare for subsequent statistical output. The PREAMBLE is similar to an extensive COMMON statement in the FORTRAN language in that everything declared in the PREAMBLE is essentially global, that is, defined for use throughout the program. This may be contrasted with variable and array definition in routines and events which insures that the variables and arrays so defined are local to the particular event or routine.

30

The PREAMBLE is also used to pack variables and arrays. Since extensive use of variable and array packing is used in STAR, this feature is discussed prior to the definition of the variables and arrays themselves. Two types of packing are utilized in STAR: field packing and bit packing.

Field packing is used to pack integer arrays such that each byte (8 bits) of an IBM 360-67 word (4 bytes or 32 bits) is used to represent an array location. Packing in this manner can reduce array core requirements to one quarter of their full word requirements. However, packing does limit the value that can be represented in an array location to $2**8 - 1$ or 255. This maximum value is more than sufficient for the packed integer arrays used by STAR. The packed arrays can be identified in the PREAMBLE by a statement of the form: THE SYSTEM OWNS A TABLE(*/4). This is followed by a statement of the form: DEFINE TABLE AS A 2-DIMENSIONAL INTEGER ARRAY.

Bit packing is used to pack attributes of temporary entities. If desired, each bit of an IBM 360-67 word may be used to represent a variable. Again, the maximum value represented by a packed variable is $2**B - 1$, where B is the number of bits used. For example, an entity might have the following packed attributes: A (NAME(1-5), AGE(6-12), ANNUAL.INCOME(13-31), SEX(32-32)). In this example, NAME may represent a maximum value of $2**5 - 1$ or 31, AGE a value of 127, ANNUAL.INCOME a value of 524287, and SEX a value of 1.

Unless otherwise stated in the discussion that follows, all data values are initialized in the MAIN routine. Following this discussion a listing of the PREAMBLE coding is provided and a brief explanation is given for PREAMBLE lines which are not straightforward.

REAL Variables

X.STOP - The X coordinate on the battlefield used to stop RED forces from advancing.

Y.STOP - Similar to X.STOP except that it represents a Y coordinate on the battlefield.

R.PCT.ATT - A percent attrition for RED forces which is used to stop the simulation. If the RED forces reach or exceed this value, the simulation will halt.

B.PCT.ATT - The BLUE force percent attrition which used in a manner similar to R.PCT.ATT above.

DEF.TIME - The amount of time a TANK remains in full defilade following a firing sequence.

CONSTANT - The cosine of the TANK's search sector angle.

AREA - The TANK's search sector angle in degrees.

B.AREA - BLUE element analog of AREA.

R.AREA - RED element analog of AREA.

MMM - The time in seconds that it takes a TANK to detect a weapon system which is in the TANK's search sector and has fired on the TANK.

NNN - The time in seconds that it takes a TANK to detect a weapon system which is not in the TANK's search sector and has fired at the TANK.

32

DELTA.T - The time interval in seconds between successive detection scheduling events (event STEP.TIME).

PCT.VIS - The percentage of a target visible to the observer.

CRITICAL.VALUE - A number between 0 and 1 which is compared to the percentage of the target visible to the observer (PCT.VIS). Line of sight does not exist if PCT.VIS is less than CRITICAL.VALUE

W.K.C. - A range, in meters, which when added to the maximum effective range of a weapon, limits the maximum detection range to the sum of these two values.

INTEGER Variables

LIN - A variable used to count the number of calls to the line of sight routine.

RC.COUNT - The number of BLUE TANKs created; used as a reference value by RED TANKs.

BC.COUNT - The number of RED TANKs created; used as a reference value by BLUE TANKs.

TTT - A variable used to count the total number of rounds fired.

LINE.OF.SIGHT.EXISTS - A variable is assigned the value of 1 if PCT.VIS is greater than CRITICAL.VALUE, and 0 otherwise.

R.NUM.ALIVE - The starting number of RED weapon systems.

B.NUM.ALIVE - The starting number of BLUE weapon systems.

BLUE - A variable representing the color of friendly forces.

RED - A variable representing the color of opposing forces.

REAL Arrays

P.V - A single element array which, when returned as an argument from FORTRAN routine LOS, contains the value of PCT.VIS.

ZH - A single element array which, when returned as an argument from FORTRAN routine ELEV, contains the value of a TANK's Z coordinate (elevation).

INTEGER Arrays

BBBPOINT - An array containing the pointer values of all BLUE TANKs.

RRRPOINT - An array containing the pointer values of all RED TANKs.

TEMP.TGT - A holding array used to store pointer values from a TANK's list of detected targets.

LIST - An array containing the pointer values of a TANK's detected targets. Each TANK has a list which is accessed by referencing the pointer value of the LIST.

TARGET - A 2-dimensional array, each row of which contains the pointer value to a TANK's LIST and the pointer value of the TANK.

DS1 - The danger state array containing BLUE engagement priorities and preferred ammunition selections within rangebands.

DS2 - The danger state array containing RED engagement priorities and preferred ammunition selections within rangebands.

ALPHA Arrays

QQ - An array used to store alphanumeric abbreviations for weapon types. This array is accessed when the results of an engagement are printed.

System Owned Sets

TANKS - The set to which all weapon systems belong.

BLUE.ALIVE - The set to which all BLUE TANKs belong initially. BLUE TANKs are removed from this set as they are killed.

RED.ALIVE - The set to which all RED TANKs belong initially. RED TANKs are removed from this set as they are killed.

Permanent Entities and Sets

COMPANY.COMMANDER - A permanent entity which is used solely as the owner of a company set, COMP.UNIT. This set contains all TANKs which are assigned to the same company.

PLATOON.LEADER - A permanent entity which is used solely as the owner of a platoon set, PLT.UNIT. This set contains all TANKs which are assigned to the same platoon.

Temporary Entities and Attributes

The basic element in STAR is the temporary entity called TANK. Each TANK has a number of characteristics called attributes which are defined below. Those attributes, which must be input by the user, are discussed in the Input Requirements chapter of this thesis.

NAME - An integer variable whose value may not exceed 1023.

COLOR - An integer variable whose value is 1 if the color is BLUE, and 0 if the color is RED.

WPN.TYPE - An integer variable representing a particular type of weapon system (e.g., XM1, BMP, IFV, etc.). The value of WPN.TYPE may not exceed 31. In the current version of STAR only 9 types of weapons are played and are numbered sequentially from 1 to 9. For a list of weapon types, see input requirements chapter or table I.

VEH.TYPE - An integer variable representing a particular lethality configuration of a weapon system. Although not currently used in STAR, its value may not exceed 31.

ALIVE.DEAD - An integer variable with a value of 0 if the TANK is alive, and a value of 1 if the TANK is dead.

HIT.STATE - An alpha variable which contains an abbreviation of the TANK's status (e.g., DEAD, HIDE, etc.).

POINTER - An integer variable which contains the pointer value assigned to the TANK upon its creation.

X.CURRENT - A real variable representing the TANK's current X coordinate on the battlefield.

Y.CURRENT - A real variable representing the TANK's current Y coordinate on the battlefield.

Z.CURRENT - A real variable representing the TANK's current Z coordinate (elevation) in meters on the battle-field.

SPD - A real variable representing the TANK's current speed in meters per second.

T.SPD - A real variable representing the simulated battle time at which a TANK's speed was last set.

DIR.OF.MVMT - A real variable representing the direction of movement of the TANK. This value is entered in mils and converted to radians in the program.

PRI.DIR - A real variable representing the primary direction of search for a TANK. This value is entered in mils and converted to radians in the program.

DEFNUM - An integer variable representing the current defilade condition of the TANK. This attribute is assigned the following values: 1 - full defilade, 2 - turret defilade, 3 - firing defilade, 4 - seeking concealment while stopping to fire, 5 - defilade condition determined by the terrain model.

AP.TOW - An integer variable representing the number of rounds of ammunition type 1 which are currently on board the TANK. For XM1s and T72s, this represents armor piercing

ammunition. For IFVs, ITVs, and BMPs, this represents
anti-tank guided missiles.

HE.DRAG - An integer variable representing the number
of rounds of ammunition type 2 which are currently on board
the TANK. For XMls and T72s, this represents high explosive
anti-tank ammunition. For DRAGONs this represents a short
range anti-tank missile.

AW1.OR.MSL3 - An integer variable representing the
number of rounds of ammunition type 3 which are currently
on board the TANK. For XMls this represents .50 caliber
ammunition counted in 10 round bursts. For T72s this
represents 12.7mm machinegun ammunition counted in 10 round
bursts. For BMPs this represents 73mm high explosive
ammunition. For IFVs this represents 25mm Bushmaster
ammunition counted in 10 round bursts. For DIVADs and ZSUs
this represents their basic air defense ammunition counted
in 10 round bursts.

AW2.OR.ADM - An integer variable representing the number
of rounds of ammunition type 4 which are currently on board
the TANK. The current version of STAR does not use this
attribute, but it is provided for the user.

HITSHOT - An integer variable representing the number of hits (catastrophic kills) scored by a TANK during its current firing sequence.

MISSSHOT - An integer variable representing the number of misses (anything other than a catastrophic kill) scored by a TANK during its current firing sequence.

PROJO - An integer variable representing the type of ammunition currently selected by the TANK.

OP.RNG - An integer variable representing the maximum range at which a TANK will attempt to engage another TANK.

MZL.VEL - An integer variable representing the velocity of the currently selected ammunition type (PROJO).

M1, M2, M3, M4 - Integer variables representing the velocities associated with each of the 4 ammunition types.

ND.HIT - An integer variable representing the number of hits sustained by a TANK which have caused no damage.

M.HIT - An integer variable representing the number of hits sustained by a TANK which have caused mobility damage.

F.HIT - An integer variable representing the number of hits sustained by a TANK which have caused firepower damage.

MF.HIT - An integer variable representing the number of hits sustained by a TANK which caused simultaneous mobility and firepower damage.

K.HIT - An integer variable representing the number of hits sustained by a TANK which were sufficient to cause a catastrophic kill (K-kill).

NUM.HIT - An integer variable representing the total number of hits, including no damage hits, sustained by a TANK.

FIRED.AT - An integer variable representing the total number of rounds fired at a TANK.

SEC, PLT, CO, BN, RGT, BDE, DIV - Integer variables representing the identification number of the section, platoon, company, battalion, regiment, brigade, and division to which the TANK is assigned.

FIP - An integer variable which is assigned the value 1 if a TANK has a firing in progress, and a 0 otherwise.

SCHED - An integer variable which is assigned the value 1 if a TANK has a fire event scheduled, and a 0 otherwise.

SECOND.SHOT - An integer variable which is assigned the value 1 if the TANK is firing a second round on the same target.

PH - A real variable representing the probability of hit as determined by routine COMPUTE.

FOE - An integer variable representing the pointer value
of the TANK's current target.

CHECK.TIME - A real variable representing the time at
which a TANK's next fire event is to occur.

RANGE - A real variable representing the range in meters
to a TANK's current target.

STEPS - An integer variable representing the number of
STEP.TIME events scheduled for the TANK.

C.1, C.2, C.3, C.4 - Integer variables representing
the amount of ammunition loaded in each of five user defined
compartments on a TANK. These attributes are not used in
the current version of STAR, but are provided for use with
vulnerability packages which are functions of ammunition
storage location.

PLOW.COND - An integer variable representing the
availability of a mineplow on a TANK. The following codes
are used: 0 = no plow available, 1 - plow available but
not in use, 2 - plow being used.

DUMMY - An integer variable which may take on values 0
or 1. This variable is not currently used in STAR.

MKILL - An integer variable assigned the value 1 if a
mobility kill is sustained by a TANK.

FKILL - An integer variable assigned the value 1 if a firepower kill is sustained by a TANK.

MFKILL - An integer variable assigned the value 1 if a mobility and firepower kill is sustained by a TANK.

KKILL - An integer variable which is assigned a value 1 if the TANK has been castrophically killed.

COCDR, PLTLDR, SECLDR - Integer variables representing the names of a TANK's company commander, platoon leader, and section leader respectively.

X.L, Y.L - Real variables which form a vector representing the left sector limit line of a TANK's search sector.

X.R, Y.R - Real variables which form a vector representing the right sector limit line of a TANK's search sector.

M.D - A real variable representing the accumulated percentage of mobility damage sustained by a TANK.

F.D - A real variable representing the accumulated percentage of firepower damage sustained by a TANK.

HTO - A real variable representing the height of the observer, in meters, above macro terrain.

HTTGT - A real variable representing the height of the target, in meters, above macro terrain.

Events

LOC.UPD - This event is used to periodically update the X, Y, and Z coordinates of TANKs on the battlefield. The pointer value of the TANK being updated is required as an argument.

STEP.TIME - This event is used to schedule detection events. The pointer value of a BLUE TANK is required as an argument.

DETECT - This event processes the detections scheduled by event STEP.TIME. The pointer values of the observer and the detected TANK are required as arguments.

TARGET.SELECT - This event uses danger state arrays, a TANK's list of detected targets, and tactical fire control modules to select targets and schedule FIRE events. The pointer value of the selecting TANK is required as an argument.

FIRE - This event simulates a TANK firing on its target. The pointer value of the firer and the target are required as arguments.

IMPACT - This event simulates the consequence of a FIRE event. The pointer values of the firer and the target are required as arguments.

FINAL.DEATH - This event changes the status of a TANK from alive to dead after an appropriate period of time. The pointer value of the TANK to be killed is required as an argument.

HIDE - This event changes the defilade condition (DEFNUM) of a TANK at the appropriate time. The pointer value of the TANK and the type of defilade change are required arguments.

ATTRITION.CHECK - This event periodically checks the total losses by RED and BLUE forces and schedules a halt to the simulation if required.

STOP.SIMULATION - This event prints final output and brings execution of STAR to a halt.

FORTRAN Routines

SETUP - This routine reads initial values for terrain features which represent the battlefield.

ELEV - This routine calculates the Z coordinate (elevation) for a TANK at a given location on the battle-field. ELEV requires three arguments: the X coordinate

of the TANK, the Y coordinate of the TANK, and the single

element array ZH, whose value upon return from ELEV is the

Z coordinate of the TANK.

LOS - This routine calculates the percentage of the

target visible to the observer. LOS requires 9 arguments:

the X, Y, and Z coordinates of the observer and target, the

heights of the observer and target, and the single element

array P.V, whose value upon return from LOS will be the

percentage of the target visible to the observer.

Coding and Brief Explanation

The coding for the PREAMBLE is shown below, followed by

a brief explanation:

1   PREAMBLE

2   DEFINE X.STOP,Y.STOP,PCT.ATT,B.PCT.ATT,DEF.TIME,

CONSTANT,AREA,

3   HALT.TIME,MMM,DELTA.T,NNN,B.AREA,R.AREA,CRITICAL.

VALUE,

4   PCT.VIS,AND W.K.C AS REAL VARIABLES

5   DEFINE LIN,RC.COUNT,BC.COUNT,TTT,LINE.OF.SIGHT.

EXISTS,R.NUM.ALIVE,

6   B.NUM.ALIVE,BLUE,AND RED AS INTEGER VARIABLES

```
 7   DEFINE QQ AS A 1-DIMENSIONAL ALPHA ARRAY

 8   DEFINE P.V AS A REAL,1-DIMENSIONAL ARRAY

 9   DEFINE ZH AS A REAL, 1-DIMENSIONAL ARRAY

10   DEFINE BBBPOINT AS AN INTEGER 1-DIMENSIONAL ARRAY

11   DEFINE RRRPOINT AS AN INTEGER 1-DIMENSIONAL ARRAY

12        DEFINE TEMP.TGT AS AN INTEGER, 1-DIMENSIONAL ARRAY

13        DEFINE TARGET AS AN INTEGER, 2-DIMENSIONAL ARRAY

14        DEFINE LIST AS AN INTEGER,  1-DIMENSIONAL ARRAY

15        DEFINE BLUE TO MEAN 1

16        DEFINE RED TO MEAN O

17   THE SYSTEM OWNS A TANKS

18   THE SYSTEM OWNS A BLUE.ALIVE AND A RED.ALIVE

19   THE SYSTEM HAS A DS1(*/4)

20   DEFINE DS1 AS A 2-DIMENSIONAL INTEGER ARRAY

21   THE SYSTEM HAS A DS2(*/4)

22   DEFINE DS2 AS A 2-DIMENSIONAL INTEGER ARRAY

23   GENERATE LIST ROUTINES

24   PERMANENT ENTITIES

25   EVERY COMPANY.COMMANDER OWNS A COMP.UNIT

26   EVERY PLATOON.LEADER OWNS A PLT.UNIT

27        TEMPORARY ENTITIES
```

28   EVERY TANK HAS A (NAME(1-10),COLOR(11-11),WPN.TYPE

(12-16),VEH.TYPE(17-21),

29   ALIVE.DEAD(22-32)),A HIT.STATE, A POINTER, A X.CURRENT,

A Y.CURRENT,

30   A Z.CURRENT, A SPD, A T.SPD, A DIR.OF.MVMT, A PRI.

DIR, A (DEFNUM(1-8),

31   AP.TOW(9-14),HE.DRAG(15-20),AW1.OR.MSL3(21-26),AW2.

OR.ADM(27-32)),

32   A (HITSHOT(1-2),MISSSHOT(3-4),PROJO(5-7),OP.RNG(8-19),

MZL.VEL(20-31)),

33   A (M1(1-16),M2(17-32),A (M3(1-16),M4(17-32)),A

(ND.HIT(1-4),N.HIT(5-8),

34   F.HIT(9-12),MF.HIT(13-16),K.HIT(17-20),NUM.HIT(21-26),

FIRED.AT(27-32)),

35   A (SEC(1-11),PLT(12-22),CO(23-32)), A (BN(1-7),

RGT(8-15),BDE(16-18),DIV(19-21),

36   FIP(22-22),SCHED(23-23),SECOND.SHOT(24-25)),A PH, A

FOE, A CHECK.TIME, A RANGE,

37   A (STEPS(1-2), C.1(3-8),C.2(9-15),C.3(16-21),C.4

(22-27),PLOW.COND(28-32)),

38   A (DUMMY(1-1),MKILL(2-11),FKILL(12-21),MFKILL(22-31),

KKILL(32-32)),

39   A COCDR, A PLTLDR, A SECLDR, A X.L, A Y.L, A X.R,

A Y.R, A M.D, A F.D, A HTO,

40   A HTTG, AND MAY BELONG TO A TANKS, A PLT.UNIT, A

COMP.UNIT, A RED.ALIVE,

41   AND A BLUE.ALIVE

42   DEFINE C.1,C.2,C.3,C.4,DEFNUM,NAME,COLOR,WPN. TYPE,

ALIVE.DEAD,FIP,SCHED,STEPS,

43   SECOND.SHOT,SEC,PLT,CO,BN,RGT,BDE,DIV,POINTER,

FOE,COCDR,PLTLDR,SECLDR,

44   ND.HIT,M.HIT,F.HIT,MF.HIT,K.HIT,NUM.HIT,FIRED.

AT,AP.TOW,HE.DRAG,

45   AW1.OR.MSL3,AW2.OR.ADM,M1,M2,M3,M4,VEH,TYPE,

MISSSHOT,HITSHOT,PROJO.

46   OP.RNG,MKILL,FKILL,MFKILL,KKILL,PLOW.COND,

MZL.VEL, AND DUMMY AS

47   INTEGER VARIABLES

48   DEFINE HIT.STATE AS AN ALPHA VARIABLE

49   EVENT NOTICES INCLUDE STOP.SIMULATION AND ATTRITION.

CHECK

50   EVERY HIDE HAS A COVER AND A CODE

51   EVERY FINAL.DEATH HAS A PURPOSE

52   EVERY LOC.UPDATE HAS A VEHICLE

48

53   EVERY STEP.TIME HAS A ENTIRE.TANK

54   EVERY DETECT HAS A WHOLE.TANK AND A DET.FOE.
ENTIRE.TANK

55   EVERY TARGET.SELECT HAS A FIRING.TANK

56   EVERY FIRE HAS A SHOOTING.TANK AND A CORE.POINTER.
OR.TGT.ID

57   EVERY IMPACT HAS A TANK.THAT.SHOT AND A BLOCK.
POINTER.OR.TGT.ID

58   DEFINE LOS AS A FORTRAN ROUTINE GIVEN 9 ARGUMENTS

59   DEFINE ELEV AS A FORTRAN ROUTINE GIVEN 3 ARGUMENTS

60   DEFINE SETUP AS A FORTRAN ROUTINE GIVEN 0 ARGUMENTS

61   TALLY LIN AS THE NUMBER OF   PCT.VIS

62   END

Lines 15-16 allow the use of the words BLUE and RED
throughout the program when referring to a TANK's color.

Lines 17-18 define sets owned by the system, as opposed
to those owned by entities.

Lines 19 and 21 declare DS1 and DS2 as packed integer
arrays.

Line 23 informs the system that the list routine option
is desired. This option allows the user to print all arrays
and contents of sets using simplified commands.

Line 25 establishes set ownership of the company sets by permanent entities called COMPANY.COMMANDERs.

Lines 40-41 allow the temporary entities, TANKs, to belong to a variety of sets.

Line 61 establishes a SIMSCRIPT statistical routine which counts the number of times that the line of sight routine is called.

B. MAIN PROGRAM

Description

The MAIN program of the STAR model prepares the model for execution. The arrays that were defined in the PREAMBLE are dimensioned to the appropriate size by the use of the RESERVE statement. The MAIN program calls the FORTRAN routine SETUP which prepares the appropriate data for the FORTRAN routines LOS and ELEV. The MAIN program sets the initial values of the variables globally defined in the PREAMBLE by use of the LET or READ statement.

A major function of the MAIN program is to create the entities, sets, and attributes defined in the PREAMBLE. Two separate DO LOOPs exist in the STAR MAIN program for creation of entities. The first creates every BLUE weapon system

to be simulated, reads or sets the initial values of the weapon system attributes and files the entity created in the appropriate sets. The second DO LOOP performs the same function for RED weapon systems in the simulation.

The final function of the MAIN program is to schedule those events which must be executed immediately upon start of the simulation.

Local Variables

The following are local variables of the MAIN program:

I - An integer variable used as a counter in the DO LOOPs which create the entities of the simulation, and as a counter in the loop which sets the initial values in the random number streams used in the simulation.

J - An integer variable used as a counter in the loop which sets the initial values in the random number streams used in the simulation.

PNUM - An integer variable read by the MAIN program which indicates the number of platoon sets (PLT.UNIT) to be created.

CNUM - An integer variable read by the MAIN program which indicates the number of company sets (COMP.UNIT) to be created.

The coding for the MAIN program is shown below, followed by a brief explanation:

```
1   MAIN
2   DEFINE I,J,PNUM,AND CNUM AS INTEGER VARIABLES
3   RESERVE DS1(*,*) AS 15 BY 24
4   RESERVE P.V(*) AS 1
5   RESERVE ZH(*) AS 1
6   RESERVE DS2(*,*) as 9 BY 40
7   RESERVE TARGET(*,*) AS 250 BY 2
8   RESERVE TEMP.TGT(*) AS 150
9   RESERVE QQ(*) AS 9
10  LET QQ(1)= "XM1"
11  LET QQ(2)= "XM1"
12  LET QQ(3)= "IFV"
13  LET QQ(4)= "ITV"
14  LET QQ(5)="DVAD"
15  LET QQ(6)="DRAG"
16  LET QQ(7)="T72"
17  LET QQ(8)="BMP"
18  LET QQ(9)="ZSU"
19  CALL SETUP
20  READ R.NUM.ALIVE, B.NUM.ALIVE, DELTA.T
```

```
21   READ PNUM, CNUM

22   LET N.PLATOON.LEADER = PNUM

23   LET N.COMPANY.COMMANDER = CNUM

24   CREATE EVERY PLATOON.LEADER

25   CREATE EVERY COMPANY.COMMANDER

26   READ   X.STOP, Y.STOP, R.PCT.ATT,    B.PCT.ATT

27   READ DS1

28   READ DS2

29   RESERVE BBBPOINT(*) AS R.NUM.ALIVE

30   RESERVE RRRPOINT(*) AS B.NUM.ALIVE

31   LET LINES.V=70

32   LET DEF.TIME = 20.0

33   LET CRITICAL.VALUE = .2

34   LET B.AREA=90.

35   LET R.AREA=90.0

36   LET AREA = B.AREA

37   LET CONSTANT = COS.F((AREA/360.0)* 2.0 * PI.C)

38   LET W.K.C = 250

39   LET NNN=2.

40   LET LIN=0

41   LET MMM=6.

42   LET RC.COUNT = 0
```

```
43  LET BC.COUNT = 0

44  LET TTT = 1

45  FOR I = 1 TO B.NUM.ALIVE, DO

46  LET RC.COUNT = RC.COUNT + 1

47  CREATE A TANK

48  READ NAME(TANK), COLOR(TANK), WPN.TYPE(TANK), VEH.
TYPE(TANK), AP.TOW(TANK),

49  HE.DRAG(TANK), AW1.OR.MSL3(TANK), AW2.OR.ADM(TANK),
SEC(TANK), PLT(TANK),

50  CO(TANK), BN(TANK), RGT(TANK), BDE(TANK), DIV(TANK),
X.CURRENT(TANK),

51  Y.CURRENT(TANK), SPD(TANK), DIR.OF.MVMT(TANK), PRI.
DIR(TANK), MI(TANK),

52  M2(TANK),M3(TANK),M4(TANK),COCDR(TANK),PLTLDR(TANK),
SECLDR(TANK)

53  , OP.RNG(TANK)

54  LET DEFNUM(TANK) = 2

55  RESERVE LIST(*) AS 1

56  LET LIST(1) = 0

57  LET HTO(TANK) = 3.0

58  LET HTTGT(TANK) = 3.0

59  LET POINTER(TANK) = TANK
```

```
60   LET RRRPOINT(I) = TANK

61   LET TARGET(I,1) = LIST(*)

62   LET LIST(*) = 0

63   LET TARGET(I,2) = TANK

64   LET DIR.OF.MVMT(TANK) = DIR.OF.MVMT(TANK) * 2 * PI.
C/6400.0

65   LET PRI.DIR(TANK)=PRI.DIR(TANK)*2*PI.C/6400.0

66   LET COCDR(TANK) = RRRPOINT(COCDR(TANK))

67   LET PLTLDR(TANK) = RRRPOINT(PLTLDR(TANK))

68   LET SECLDR(TANK) = RRRPOINT(SECLDR(TANK))

69        FILE TANK IN TANKS

70   FILE TANK IN BLUE.ALIVE

71   FILE TANK IN PLT.UNIT(PLT(TANK))

72   FILE TANK IN COMP.UNIT(CO(TANK))

73   LOOP

74   FOR I = B.NUM.ALIVE+1 TO B.NUM.ALIVE+R.NUM.ALIVE, DO

75   LET BC.COUNT = BC.COUNT + 1

76   CREATE A TANK

77   READ NAME(TANK), COLOR(TANK), WPN.TYPE(TANK), VEH.
TYPE(TANK), AP.TOW(TANK),

78   HE.DRAG(TANK), AW1.OR.MSL3(TANK), AW2.OR.ADM(TANK),
SEC(TANK), PLT(TANK),
```

```
79  CO(TANK), BN(TANK), RGT(TANK), BDE(TANK), DIV(TANK),
X.CURRENT(TANK),

80  Y.CURRENT(TANK), SPD(TANK), DIR.OF.MVMT(TANK), PRI.
DIR(TANK), M1(TANK),

81  M2(TANK),M3(TANK),M4(TANK),COCDR(TANK),PLTLDR(TANK),
SECLDR(TANK)

82  , OP.RNG(TANK)

83  LET DEFNUM(TANK) = 5

84  RESERVE LIST(*) AS 1

85  LET LIST(1) = 0

86  LET HTO(TANK) = 3.0

87  LET HTTGT(TANK) = 3.0

88  LET POINTER(TANK) = TANK

89  LET BBBPOINT(I-B.NUM.ALIVE) = TANK

90  LET TARGET(I,1) = LIST(*)

91  LET LIST(*) = 0

92  LET TARGET(I,2) = TANK

93  LET SPD(TANK) = 5.0

94  LET DIR.OF.MVMT(TANK) = DIR.OF.MVMT(TANK) * 2 * PI.
C/6400.0

95  LET PRI.DIR(TANK)=PRI.DIR(TANK)*2*PI.C/6400.0

96  LET COCDR(TANK)=BBBPOINT(COCDR(TANK)-B.NUM.ALIVE)
```

97   LET PLTLDR(TANK)=BBBPOINT(PLTLDR(TANK)-B.NUM.ALIVE)

98   FILE TANK IN TANKS

99   FILE TANK IN RED.ALIVE

100 FILE TANK IN PLT.UNIT(PLT(TANK))

101 FILE TANK IN COMP.UNIT(CO(TANK))

102 LOOP

103 FOR EACH TANK IN BLUE.ALIVE, DO

104 CALL ELEV GIVEN X.CURRENT(TANK), Y.CURRENT(TANK),
ZH(*)

105 IF ZH(1) LE 250.0 LET ZH(1) = 250.0 ALWAYS LET
Z.CURRENT(TANK) = ZH(1)

106 LOOP

107 FOR I = 1 TO 9, FOR J = 1 TO 10, LET XX=UNIFORM.
F(0.,1.,I)

108 FOR EACH TANK IN TANKS LET PRI.DIR(TANK) = PRI.DIR
(TANK) + DIR.OF.MVMT(TANK)

109 FOR EACH TANK IN TANKS CALL SET.SECTOR(TANK)

110 FOR EACH TANK IN RED.ALIVE SCHEDULE A LOC.UPDATE
(TANK) NOW

111 FOR EACH TANK IN BLUE.ALIVE LET STEPS(TANK) = 1

112 PRINT 1 LINE THUS
RND FIR TYPE AM TGT TYPE TIME.V STAT RANGE X-FIR Y-FIR X-TGT
Y-TGT   57

113 FOR EACH TANK IN BLUE.ALIVE SCHEDULE A STEP.TIME(TANK)
NOW

114 SCHEDULE AN ATTRITION.CHECK IN 10.0 * DELTA.T UNITS

115 START SIMULATION

116 STOP

117 END

Lines 1-2 define the MAIN program and local variables.

Lines 3-8 dimension the numeric arrays defined in the
PREAMBLE.

Lines 9-18 dimension the alphanumeric array QQ and place
the appropriate alphanumeric characters in the array
locations.

Line 19 calls the FORTRAN routine SETUP.

Line 20 reads the initial number of RED and BLUE systems
in the simulation and the time increment for the simulation.

Line 21 reads the number of platoon and company sets
to be created.

Lines 22-25 create the platoon leader and company commander
entities and simultaneously create subscripted company and
platoon sets.

Line 26 reads the X and Y coordinates used for stopping criteria, and reads the percent attrition for BLUE and RED systems to be used as a stopping criteria.

Lines 27-28 read in values of the danger state arrays.

Lines 29-30 dimension the arrays BBBPOINT and RRRPOINT.

Lines 31-44 set initial values for PREAMBLE defined global variables.

Lines 45-53 create each entity representing the BLUE weapon systems and read in initial attribute values.

Lines 54-63 set initial attribute values of the entities created, initialize the target list for each entity created (LIST) and place the appropriate pointer values in the array TARGET.

Lines 64-65 convert the value of PRI.DIR and DIR.OF.MVMT attributes from mils to radians

Lines 66-68 convert the attributes COCDR, PLTLDR, and SECLDR (which were read in as vehicle numbers) to the proper pointer value of the entity's company commander, platoon leader and section leader.

Lines 69-73 place the entity created in the appropriate sets and loop back to line 45.

Lines 74-82 create each entity representing RED weapon systems, and read in the initial attribute values.

Lines 83-93 set initial values of the attributes of the entities created, initialize the target list for each entity (LIST) and place the appropriate pointer values in the array TARGET.

Lines 94-95 convert the PRI.DIR and DIR.OF.MVMT attributes to radians.

Lines 96-97 convert the COCDR, and PLTLDR attributes to the proper pointer value of the entity's platoon leader or company commander.

Lines 98-102 file the entity in the proper sets and loop back to line 74.

Lines 103-106 set initial elevation for the BLUE elements by calling the FORTRAN routine ELEV.

Line 107 sets the starting values in each of the random number streams used in the simulation.

Line 108 sets the PRI.DIR attribute of each entity to the form used in routine SET.SECTOR.

Line 109 calls routine SET.SECTOR for every entity, which sets the limits of the entity's search sector.

Line 110 schedules event LOC.UPDATE to occur for every RED weapon system immediately upon start of the simulation.

Line 111 sets the STEPS attribute for each BLUE entity.

Line 112 prints the column headings for printed output.

Line 113 schedules event STEP.TIME to occur for every BLUE weapon system immediately upon start of the simulation.

Line 114 schedules event ATTRITION.CHECK to occur at a time ten times DELTA.T into the simulation.

Line 115 starts the simulation.

Lines 116-117 are the required MAIN program control statements.

C. EVENT LOC.UPDATE

Description

Event LOC.UPDATE is called every DELTA.T units by the system timer and is used to update the X, Y, and Z coordinates of each RED TANK that is currently alive. This event insures that all positions have been updated prior to the execution of event STEP.TIME.

Local Variable

The following is a local variable of this routine:

TANK - An integer variable containing the pointer value

of the weapon system for which a location update is required.

Coding and Brief Explanation

The coding for event LOC.UPDATE is shown below,

followed by a brief explanation:

```
1   UPON LOC.UPDATE(TANK)

2   DEFINE TANK AS AN INTEGER VARIABLE

3   IF ALIVE.DEAD(TANK) EQ 1 RETURN ELSE

4   CALL LOC(TANK)

5   SCHEDULE A LOC.UPDATE(TANK) IN DELTA.T UNITS

6   RETURN END
```

Lines 1-2 define the event and local variable.

Line 3 tests the ALIVE.DEAD attribute of the TANK to

prevent dead TANKs from having their locations updated.

Line 4 calls routine LOC which calculates new X, Y, and

Z coordinates for the TANK.

Lines 5-6 reschedule another LOC.UPDATE for this TANK

in DELTA.T units (currently 30 seconds). Control is then

returned to the system timer.

D. EVENT STEP.TIME

Description

   Event STEP.TIME is used by STAR to generate detections
of battlefield elements by one another. STEP.TIME is
scheduled to occur every DELTA.T units (currently 30 seconds)
and affects all TANKs which are alive at the beginning of
the time period. Prior to STEP.TIME, each TANK's X, Y, and
Z coordinates are updated by event LOC.UPDATE.

   Each BLUE TANK that enters event STEP.TIME has the
opportunity to schedule a detection on a RED TANK. Like-
wise, each RED TANK has the opportunity to schedule a de-
tection on the BLUE TANK that generated the STEP.TIME
event. Detection times are generated by routine CARDIO
which is discussed in a subsequent section.

   Range, sector, and time filters are used by event
STEP.TIME to prevent unnecessary line of sight calls.
Although the current line of sight algorithm (FORTRAN
routine LOS) is very efficient, it still requires many
calculations. Consequently, the filters mentioned above
and described in the coding section below are important in
reducing execution time.

Local Variables

The following are local variables of this event:

A - An integer variable containing the pointer value of the BLUE TANK which generated the current STEP.TIME event. Note that the global variable TANK represents RED TANKs in this event.

ANSWER - An integer variable which is assigned a value of 1 if a target is in the observor's search sector, and 0 otherwise.

LOSE - An integer variable which is assigned a value of 1 if line of sight exists between two elements, and 0 otherwise.

R - A real variable representing the range in meters between two TANKs.

RN.B - A real variable assigned a random value from a Uniform (0,1) probability distribution. This value is passed as an argument to routine CARDIO.

RN.R - A real variable with the same characteristics as RN.R.

BDET.TIME - A real variable, which when returned from routine CARDIO, represents the time required to detect a RED TANK by a BLUE TANK.

64

RDET.TIME - Similar to BDET.TIME except that it is used
for RED TANKs detecting BLUE TANKs.

<u>Coding and Brief Explanation</u>

The coding for event STEP.TIME is shown below, followed
by a brief explanation:

```
 1  UPON STEP.TIME(A)

 2  DEFINE ANSWER AS AN INTEGER VARIABLE

 3  DEFINE RN.B, RN.R, AND R AS REAL VARIABLES

 4  DEFINE LOSE AND A AS INTEGER VARIABLES

 5  DEFINE RDET.TIME AND BDET.TIME AS REAL VARIABLES

 6  LET STEPS(A)=STEPS(A)-1

 7  LET BDET.TIME=99  LET RDET.TIME = 99

 8  IF ALIVE.DEAD(A) EQ 1 RETURN ELSE

 9  IF TIME.V GE 2.0 * DELTA.T

10  LET LIST(*) = TARGET(NAME(A),1)

11  IF DEFNUM(A) EQ 1 RELEASE LIST(*)  RESERVE LIST(*)
AS 1

12  LET TARGET(NAME(A),1) = LIST(1)=0  LET LIST(*) = 0
GO TO NORMAL

13        ELSE

14  IF LIST(1) = 0

15  CALL CHG.SEC.SEARCH(A)
```

65

```
16  ALWAYS

17  LET LIST(*)=0

18  REGARDLESS

19  'NORMAL'

20  FOR EACH TANK IN RED.ALIVE, DO

21  IF N.BLUE.ALIVE EQ A AND TIME.V GE 2.0 * DELTA.T

22  LET LIST(*) = TARGET(NAME(TANK),1)

23  IF LIST(1) = 0

24  CALL CHG.SEC.SEARCH(TANK)

25  ALWAYS

26  LET LIST(*)=0

27  ALWAYS

28  LET RN.B= UNIFORM.F(0.,1.,7)

29  LET RN.R= UNIFORM.F(0.,1.,7)

30  IF (Y.CURRENT(TANK) + X.CURRENT(TANK)/2.0) LE 8000.0

31  LET ALIVE.DEAD(TANK) = 1 REMOVE TANK FROM RED.ALIVE
GO TO LOOP ELSE

32  CALL DIST(X.CURRENT(A),Y.CURRENT(A),X.CURRENT(TANK),
Y.CURRENT(TANK)) YIELDING R

33  IF OP.RNG(A) + W.K.C. GT R  CALL SECTOR.CHECK(A,TANK)
YIELDING ANSWER
```

66

34   IF ANSWER EQ 1 CALL CARDIO(A,TANK,R,1.0,RN.B)
YIELDING BDET.TIME

35   ELSE ALWAYS

36   ALWAYS

37   IF OP.RNG(TANK) + W.K.C. GT R CALL SECTOR.CHECK(TANK,
A) YIELDING ANSWER

38   IF ANSWER EQ 1 CALL CARDIO(TANK,A,R,1.0,RN.R)
YIELDING RDET.TIME

39   ELSE ALWAYS

40   ALWAYS

41   IF BDET.TIME LT DELTA.T OR RDET.TIME LT DELTA.T

42   CALL LOS GIVEN X.CURRENT(A),Y.CURRENT(A),Z.CURRENT
(A),HTO(A),X.CURRENT(TANK),

43   Y.CURRENT(TANK),Z.CURRENT(TANK),HTTGT(TANK),P.V(*)

44   LET PCT.VIS=P.V(1)

45   IF PCT.VIS GT CRITICAL.VALUE LET LOSE=1 JUMP AHEAD
ELSE LET LOSE=0 HERE

46   'SCHEDULE' IF LOSE EQ 1 AND BDET.TIME LT DELTA.T

47   CALL CARDIO(A,TANK,R,PCT.VIS,RN.B) YIELDING BDET.TIME

48   IF BDET.TIME GT DELTA.T JUMP AHEAD ELSE

49   SCHEDULE A DETECT(A,TANK) IN BDET.TIME UNITS

50   HERE

51 REGARDLESS

52 IF LOSE EQ 1 AND RDET.TIME LT DELTA.T

53 CALL CARDIO(TANK,A,R,PCT.VIS,RN.R) YIELDING RDET.TIME

54 IF RDET.TIME GT DELTA.T JUMP AHEAD ELSE

55 SCHEDULE A DETECT(TANK,A) IN RDET.TIME UNITS

56 HERE

57 REGARDLESS

58 ALWAYS 'LOOP' LOOP

59 .IF STEPS(A) GE 2 RETURN

60 ELSE SCHEDULE A STEP.TIME(A) IN DELTA.T UNITS LET
STEPS(A)=STEPS(A)+1 RETURN END

Lines 1-5 define the event and local variables.

Line 6 decrements the steps attribute for the BLUE TANK
that generated the current STEP.TIME.

Line 7 assigns default values to BDET.TIME and RDET.TIME.

Line 8 insures that the BLUE TANK is still alive prior
to proceeding.

Lines 9-19 perform several tests after simulated time
has reached 60 seconds. If a BLUE TANK is in full defilade,
his detected list is purged. This simulates the loss of
information which accompanies being in a position from which
line of sight does not exist. If the BLUE TANK is not in

68

full defilade, but his detected list is empty, his sector of search is changed in attempt to acquire new targets in a different search area. BLUE TANKs then attempt to schedule DETECT events in the usual manner.

Line 20 effects sequential access to the RED TANKs which are still alive.

Lines 21-27 change the search sector for those RED TANKs whose detected lists are empty. Note that this is accomplished only when the first BLUE TANK accesses the set of RED TANKs.

Lines 28-29 draw random numbers for use by routine CARDIO.

Lines 30-31 prevent RED TANKs from moving off the map. By setting the ALIVE.DEAD attribute to 1 when a RED TANK reaches the stopping point, the simulation is prevented from executing indefinitely since the RED attrition percentage will eventually reach 100 percent.

Line 32 calculates the current range between two TANKs.

Lines 33-36 insure for BLUE TANKs that a potential detected target is within range and within sector. If this is the case, routine CARDIO is called with 100 percent visibility assumed. This generates a best case detection time.

Lines 37-40 perform the above tests for RED TANKs detecting BLUE TANKs.

Lines 41-45 call FORTRAN routine LOS if either potential detect time calculated above was less than DELTA.T seconds. Line of sight exists if the percentage of the target visible to the observer is greater than a user defined value (currently .2).

Line 46 insures that line of sight exists between the BLUE observer and the RED target, and that the potential detection time is less than DELTA.T seconds prior to calling routine CARDIO again.

Line 47 calls routine CARDIO and returns with a detection time for BLUE detecting RED.

Lines 48-50 schedule a detection in the amount of time returned by routine CARDIO if this time is less than DELTA.T seconds.

Lines 51-57 are similar to lines 46-50 except that they apply to RED TANKs observing BLUE TANKs.

Lines 59-60 schedule another STEP.TIME event in DELTA.T seconds if the BLUE TANK has fewer than 2 STEP.TIME events scheduled. Control is then returned to the system timer.

E. EVENT DETECT

Description

    Event DETECT is scheduled by events STEP.TIME and IMPACT. Normal detections using routine CARDIO are scheduled by event STEP.TIME, while detections as a result of being fired upon are scheduled by event IMPACT. Depending on the defilade conditions of the observer and target, line of sight conditions between these two elements and proximity of other elements to the detected threat, detected elements may be added to the observer's detected list, target selections may be scheduled or near vicinity detections may be effected.

Local Variables

    The following are local variables of this routine:

    A - An integer variable containing the pointer value of the observer.

    B - An integer variable containing the pointer value of the target.

Coding and Brief Explanation

    The coding for event DETECT is shown below, followed by a brief explanation.

    1 UPON DETECT(A,B)

    2 DEFINE A,B AS INTEGER VARIABLES

71

```
 3  DEFINE J AS AN INTEGER VARIABLE

 4  IF ALIVE.DEAD(A) EQ 1

 5  RETURN

 6  ELSE

 7  IF DEFNUM(B) EQ  1  CALL LIST.UPDATE(0,A,B,0)
RETURN  ELSE

 8  CALL LOC(A)  CALL LOC(B)

 9  CALL LOS GIVEN X.CURRENT(A),Y.CURRENT(A),Z.CURRENT
(A),HTO(A),X.CURRENT(B),

10  Y.CURRENT(B),Z.CURRENT(B),HTTGT(B),P.V.(*)

11  LET PCT.VIS=P.V(1)

12  IF PCT.VIS GT CRITICAL.VALUE LET LINE.OF.SIGHT.
EXISTS=1 JUMP AHEAD ELSE

13  LET LINE.OF.SIGHT.EXISTS=0  HERE

14  IF DEFNUM(A) EQ 1 CALL LIST.UPDATE(LINE.OF.SIGHT.
EXISTS,A,B,0) RETURN ELSE

15  IF LINE.OF.SIGHT.EXISTS EQ 1 CALL PROXIMITY.DETECT
(A,B) ALWAYS

16  RETURN

17  END
```

Lines 1-2 define the routine and local variables.

Lines 3-5 insure that the observer is still alive before proceeding.

Line 6 tests the defilade condition of the target. If the target is in full defilade, (that is, he cannot be seen), then the observer is considered to have lost detection on this target and the target, if currently in the detected list, is removed from the list. The first 0 argument in the call to routine LIST.UPDATE accomplishes this, while the second 0 argument in this call prevents a target selection from being scheduled.

Line 7 updates the X, Y, and Z coordinates of the observer and target.

Lines 8-12 determine if line of sight exists between the observer and target.

Line 13 tests the defilade condition of the observer. If the observer is in full defilade, the detection may still occur if line of sight exists and the target is still alive.

Lines 14-15 allow for the possibility of near vicinity detections if line of sight exists and the observer is not in defilade. Control is then returned to the system timer.

F.   EVENT TARGET.SELECT

Description

Event TARGET.SELECT is used by all weapon systems to
select targets on the basis of highest priority within
range bands. For some weapons systems, the priority criteria
may be altered as a result of actions by other unit members.
These actions are represented in several tactical target
selection modules. The current version of STAR uses tactical
target selection modules for the following weapon systems:
XM1, IFV, ITV, T72, and BMP. These modules are presented
in detail in later sections. In general, all weapon
systems using these modules attempt to select the highest
priority target that is not being engaged by another unit
member. IFVs, ITVs, and BMPs will not fire if all targets
in their detected list are being engaged by other unit
members. In contrast, XM1s will always engage a target from
their detected list assuming that the range and line of
sight filters are satisfied. An XM1's engagement will
be the highest priority threat not being engaged by another
unit member or, if all threats are being engaged, the target
selected will be the highest priority threat in the detected

list. T72s always attempt to acquire the FOE of their

platoon leader. Failing this, a T72 will select its

highest priority threat.

If, after a target is selected, line of sight does not

exist, the target is in full defilade or the target has

already been killed, the threat will be removed from the

firer's detected list and a re-selection will be attempted.

If all potential targets are outside the maximum

engagement range of the firer, control is returned to the

system timer.

If a selection is accomplished, event FIRE is scheduled

in the maximum of the time it takes to lay (aim) the weapon

and load (prepare to fire) the weapon.

Two points should be clarified with respect to the

variable ANSWER. An ANSWER of 0 from routine T72.TACTICS

indicates that the T72 has selected his platoon leader's

FOE. An answer of 1 indicates the converse. Within the

loop that sequentially accesses detected threats (lines

27-47), an ANSWER of 0 means that the currently accessed

threat is still a candidate for selection. An ANSWER of 1

causes this threat to be bypassed. In most cases the latter

result occurs when a threat is being engaged by another unit member.

Event TARGET.SELECT is relatively complicated and a cursory look at the tactical target selection modules prior to reading this event is recommended.

Local Variables

The following are local variables of this event:

A - An integer variable containing the pointer value of the selecting TANK.

ID - An integer variable containing the pointer value of the selected TANK.

I - An integer counter.

ANSWER - An integer variable whose value is the result of a call to a tactical target selection module. See the Description section of this event for details.

P - An integer variable which reflects the priority of the currently accessed threat in the detected list.

OLDP - An integer variable which reflects the priority of the current candidate for selection.

RND - An integer variable which contains a coded ammunition type. This ammunition is the type desired for use against the current candidate for selection.

ENGAGED - An integer variable which takes on the value 0 or 1. Used exclusively for XM1 target selection, if ENGAGED is changed to 0 as a result of a call to routine XM1.TACTICS, only subsequent targets which are not being engaged will be candidates for selection. If ENGAGED remains 1, the XM1 will select the highest priority threat from his list, regardless of its engagement status.

WHOCALLED - An integer variable which is assigned the value 1 in event TARGET.SELECT. As an argument to routine LIST.UPDATE, a value of 1 for WHOCALLED prevents another TARGET.SELECT from occurring as a result of the detected list being updated.

R - A real variable representing the range in meters between the selecting TANK and the currently accessed threat.

OLD.RANGE - A real variable representing the range in meters between the selecting TANK and the most recent candidate for selection.

Coding and Brief Explanation

The coding for event TARGET.SELECT is shown below, followed by a brief explanation:

77

```
 1  UPON TARGET.SELECT(A)

 2  DEFINE R, AND OLD.RANGE AS REAL VARIABLES

 3  DEFINE ANSWER, OLDP, P, RND, AIM, A, ID, I, ENGAGED,
AND WHOCALLED AS INTEGER

 4  VARIABLES

 5  IF DEFNUM(A) = 1 RETURN ELSE

 6  IF ALIVE.DEAD(A) EQ 1 RETURN

 7  ELSE

 8  IF FIP(A) EQ 1 RETURN

 9  ELSE

10  IF SCHED(A)=1

11  RETURN

12  ELSE

13  IF WPN.TYPE(A) EQ 7 CALL T72.TACTICS(A) YIELDING
ANSWER

14  IF ANSWER EQ 0 RETURN ELSE

15  ALWAYS

16  'SELECT'

17  CALL LOC(A)

18  'TRY.AGAIN'

19  LET LIST(*)=TARGET(NAME(A),1)

20  IF LIST(1) EQ 0
```

```
         LET
21  LIST LIST(*) = 0

22  RETURN

23  ELSE

24  LET ENGAGED = 1 LET WHOCALLED = 1 LET OLDP = 99
LET OLD.RANGE = 99999

25  FOR I=1 TO DIM.F(LIST(*)), DO

26  CALL LOC(LIST(I))

27  CALL DIST(X.CURRENT(A),Y.CURRENT(A),X.CURRENT
(POINTER(LIST(I))),

28       Y.CURRENT(POINTER(LIST(I)))) YIELDING R

29  LET RANGE(A) = R

30  GO TO XM1, XM1, IFV, ITV, DIVAD, DRAGON, T72, BMP,
ZSU PER WPN.TYPE(A)

31  'IFV'  CALL IFV.TACTICS(A,LIST(I)) YIELDING ANSWER
GO TO NEXT

32  'ITV'  CALL ITV.TACTICS(A,LIST(I)) YIELDING ANSWER
GO TO NEXT

33  'DIVAD' LET ANSWER = 0 GO TO NEXT

34  'DRAGON' LET ANSWER = 0 GO TO NEXT

35  'T72' LET ANSWER = 0 GO TO NEXT

36  'BMP'  CALL BMP.TACTICS(A,LIST(I)) YIELDING ANSWER
GO TO NEXT
```

```
37  'ZSU'  LET ANSWER = 0 GO TO NEXT

28  'NEXT' IF ANSWER EQ 1 GO TO LOOP ELSE GO TO NORMAL.
SELECT

39  'XM1' CALL XM1.TACTICS(A,LIST(I))  YIELDING ANSWER

40  'NORMAL.SELECT' CALL PRIORITY.AND.ROUND.SELECT(A,
LIST(I)) YIELDING P, RND

41  IF P EQ 99 GO TO LOOP ELSE

42  IF ANSWER LT ENGAGED GO TO REPLACE ELSE

43  IF (( P LT OLDP) OR (P EQ OLDP AND R LE OLD.RANGE))
AND (ANSWER EQ ENGAGED)

44  'REPLACE' LET OLDP=P LET OLD.RANGE=R LET PROJO(A)=
RND LET ID=LIST(I)

45  LET ENGAGED = ANSWER REGARDLESS 'LOOP' LOOP

46  LET LIST(*) = 0

47  IF OLD.RANGE LE OP.RNG(A)

48  JUMP AHEAD

49  ELSE

50  RETURN

51  HERE

52  CALL LOS GIVEN X.CURRENT(A),Y.CURRENT(A),Z.CURRENT
(A),HTO(A),
```

```
53        X.CURRENT(ID),Y.CURRENT(ID),Z.CURRENT(ID),
HTTGT(ID),P.V(*)

54  LET PCT.VIS=P.V(1)

55  IF PCT.VIS GT CRITICAL.VALUE   LET LINE.OF.SIGHT.
EXISTS=1

56  JUMP AHEAD ELSE

57  LET LINE.OF.SIGHT.EXISTS=0   HERE

58  IF LINE.OF.SIGHT.EXISTS   EQ 0

59  OR ALIVE.DEAD(ID) EQ 1

60  OR DEFNUM(ID) EQ 1

61  CALL LIST.UPDATE(0,A,ID,WHOCALLED)

62      GO TO TRY.AGAIN

63  ELSE

64  LET RANGE(A) = OLD.RANGE

65  LET FOE(A)=ID

66  LET SCHED(A)=1

67  SCHEDULE A FIRE(A,ID) IN MAX.F(UNIFORM.F(10.,20.,3),
UNIFORM.F(8.,16.,4))

68      UNITS

69  LET CHECK.TIME(A) = TIME.A(FIRE)

70  RETURN

71  END
```

Lines 1-4 define the event and local variables.

Lines 5-11 return control to the system timer if the selecting TANK is in full defilade, is dead, has a firing in progress, or has a fire scheduled.

Lines 12-16 apply only to weapons of type T72. The T72 attempts to acquire his platoon leader's FOE and schedule a FIRE event on this target. If this occurs, control is returned to the system timer. Otherwise, the T72 selects his highest priority threat in a manner similar to other weapon systems.

Lines 17-23 update the X, Y, and Z coordinates of the selecting TANK and access the TANK's list of detected threats. If the list contains no potential targets, control returns to the system timer. Otherwise, execution continues.

Line 24 sets default values for local variables.

Line 25 sequentially accesses elements in the TANK's list of detected threats.

Lines 26-29 update the X, Y, and Z coordinates of the currently accessed threat. The range between the TANK and this threat is calculated and the TANK's RANGE attribute is set to this value.

Line 30 transfers control to the appropriate label based on the weapon type of the TANK.

Lines 31-37 call a tactical target selection module, if required, and transfer control to the label 'NEXT'.

Line 38 tests the ANSWER returned above and transfers control to label 'LOOP' if ANSWER = 1, or to label 'NORMAL. SELECT' if ANSWER = 0.

Line 39 calls the tactical routine for XM1s.

Line 40 returns a priority and desired ammunition type to be used against the current threat.

Line 41 transfers control to label 'LOOP' if the priority returned is equal to 99. This indicates that the target cannot be engaged.

Line 42 allow XM1s to give higher priority to targets which are not being engaged. Any target not being engaged can replace a target which is being engaged.

Line 43 tests the values of P, R, and ANSWER against the values of OLDP, OLD.RANGE, and ENGAGED. To replace a threat as the current candidate for selection, the target must have a higher priority or, if it has the same priority, must be closer in range. Moreover, it must have the same engage-ment status as the current threat. This last condition

is always satisfied for weapons other than XMls. If the
test fails, control transfers to label 'LOOP'.

Lines 44-45 set local variables to values associated
with the new threat. If the threat list has not been
exhausted, the cycle begins again with line 27.

Line 46 releases the list of detected threats.

Lines 47-51 insure that the range to the selected
threat is within the opening range of the TANK. If not,
control returns to the system timer. Otherwise, execution
continues.

Lines 52-57 calculate the percentage of the target
visible to the selecting TANK. Line of sight is set to 1
if the percentage returned exceeds a user defined value.

Lines 58-62 remove the currently selected threat from
the list and allow another selection (label 'TRY AGAIN')
if the selected threat is dead, is in full defilade or
cannot be seen.

Lines 63-66 set the TANK's RANGE attribute to the current
range to the target, set the FOE attribute to the pointer
value of the target, and set the scheduled to fire attribute,
SCHED, to 1.

Lines 67-70 schedule a FIRE event on the selected target, set the TANK's CHECK.TIME attribute to the time for which the FIRE event is scheduled to occur, and return control to the system timer.

G. EVENT FIRE

Description

Event FIRE may be scheduled by event TARGET.SELECT, routine T72.TACTICS, and routine WE.MISS. Event FIRE is scheduled in the maximum of two times: the time it takes to aim the weapon (lay time), and the time it takes to prepare the weapon for firing (load time). For second round shots at the same target after a miss, only load time is considered. The weapon is assumed to be laid on the target.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value of the firing TANK.

ID - An integer variable containing the pointer value of the target TANK.

STOPCOUNT - An integer variable with a value of 1 when used in event FIRE. This value is used as an argument in routine STOP.TO.FIRE to indicate that the weapon system must stop prior to firing.

Coding and Brief Explanation

The coding for event FIRE is shown below, followed by a brief explanation:

```
 1  UPON FIRE(A,ID)
 2  DEFINE STOPCOUNT AS AN INTEGER VARIABLE
 3       DEFINE A AS AN INTEGER VARIABLE
 4  DEFINE ID AS AN INTEGER VARIABLE
 5       DEFINE R AS A REAL VARIABLE
 6  LET SCHED(A)=0
 7  LET STOPCOUNT=1
 8  IF DEFNUM(A) EQ 1 LET FOE(A) = 0 RETURN ELSE
 9  IF DEFNUM(ID) EQ 1 GO TO RE.SELECT ELSE
10  CALL LOC(A)   CALL LOC(ID)
11                   IF FOE(A) NE ID RETURN
12                      ELSE
13  IF CHECK.TIME(A) NE TIME.V RETURN ELSE
14  IF FIP(A) EQ 1 RETURN ELSE
15  IF ALIVE.DEAD(A) EQ 1 RETURN
```

```
16  ELSE

17  IF ALIVE.DEAD(ID) EQ 1

18  'RE.SELECT'

19  LET FOE(A) = 0

20  LET SECOND.SHOT(A) = 0

21                              LET FIP(A) = 0

22  SCHEDULE A TARGET.SELECT(A) NOW

23  RETURN

24          ELSE

25  CALL DIST(X.CURRENT(A),Y.CURRENT(A),X.CURRENT
(POINTER(ID)),

26      Y.CURRENT(POINTER(ID))) YIELDING R

27  IF R GT OP.RNG(A)

28  LET FIP(A)=0

29  LET FOE(A)=0

30  SCHEDULE A TARGET.SELECT(A) NOW

31  RETURN

32  ELSE

33  CALL LOS GIVEN X.CURRENT(A),Y.CURRENT(A),Z.CURRENT
(A), HTO(A), X.CURRENT(ID),

34  Y.CURRENT(ID), Z.CURRENT(ID), HTTGT(ID), P.V(*)
```

```
35  LET PCT.VIS=P.V(1)

36  IF PCT.VIS LT CRITICAL.VALUE

37  GO TO RE.SELECT

38  ELSE

39  IF COLOR(A)  EQ BLUE LET DEFNUM(A) = 3  ALWAYS

40  LET FIP(A) = 1

41  LET RANGE(A) = R

42  IF WPN.TYPE(A) EQ 8 OR WPN.TYPE(A) EQ 3 OR WPN.
TYPE(A) EQ 4

43  CALL STOP.TO.FIRE(A,STOPCOUNT)

44  ALWAYS

45  CALL SET.MUZZLE.VEL(A)

46  SCHEDULE AN IMPACT(A,ID) IN R/MZL.VEL(A) UNITS

47  RETURN

48  END
```

Lines 1-6 define the routine and local variables.

Line 7 resets the scheduled to fire attribute to 0.

Line 8 assigns STOPCOUNT a value of 1.

Line 9 tests to insure that the firer is not in full
defilade before proceeding.

Line 10 tests to insure that the target is not in full defilade before proceeding. If the target is in full defilade, flow is transferred to a point where another tank selection can be made (see label 'RE.SELECT').

Line 11 updates the X, Y, and Z coordinates of the firer and target.

Lines 12-13 insure that the current FOE is the one for which this FIRE event was scheduled.

Line 14 insures that this is the correct FIRE event by matching elapsed simulated time with the time that this tank's last fire was scheduled to occur.

Line 15 prevents this FIRE if the firer already has a firing in progress.

Lines 16-17 prevent this FIRE if the firer is dead.

Lines 18-25 prevent this FIRE if the target is already dead. If this is the case, the firer's FOE, SECOND.SHOT, and firing in progress attributes are reset to 0. An immediate target selection is scheduled, and control is returned to the system timer.

Lines 26-27 calculate the current distance in meters between the target and firer.

Lines 28-31 insure that the target is within the maximum
opening range of the firer. If this is not the case, the
FIP and FOE attributes are reset to 0, and control is
returned to the system timer.

Lines 32-35 calculate the percentage of the target
that is visible to the firer.

Lines 36-37 return control to the system timer if the
target's percentage visible is less than the user defined
value.

Lines 38-39 place BLUE firers in firing defilade.

Lines 40-41 set the firing in progress attribute to 1
and the range attribute to the current range between firer
and target.

Lines 42-43 halt missile firing weapon systems prior
to firing.

Lines 44-47 calculate the muzzle velocity for the type
of ammunition being fired and schedule an IMPACT event in
an amount of time equal to the time of flight of the round.
Control is then returned to the system timer.

H. EVENT IMPACT

Description

Event IMPACT is scheduled by event FIRE. Once a round
is fired, event IMPACT is always allowed to occur. If the
firer scores a catastrophic kill (K-kill), the target's
HIT.STATE attribute is changed to "DEAD", the ALIVE.DEAD
attribute is set to 1, the target is removed from the firer's
detected list, and the appropriate hit counters are incre-
mented for the target. Missile firing weapons are
restarted from their firing positions and routine WE.HIT
determines the next action for the firer.

Anything other than a catastrophic kill causes the MISS
section of event IMPACT to be activated. A detection of
the firer is scheduled in an amount of time dependent on
the firer's location with respect to the target. Missile
firing weapons are moved from their firing positions,
routine WE.MISS determines the next action for the firer and
the appropriate hit counters are incremented for the target.

Results of the IMPACT event are then printed and control
is returned to the system timer.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value of the firer.

ID - An integer variable containing the pointer value of the target.

WHOCALLED - An integer variable set equal to 10 in event IMPACT which allows a target selection to occur when the detected list is updated by the removal of a dead target.

STOPCOUNT - An integer variable set equal to 2 in event IMPACT which restarts missile firing weapon systems when routine STOP.TO.FIRE is called.

DT - A real variable used to set the time of detection of the firer by the target. DT assumes one of two values input by the user. These values are dependent on whether or not the firer is in the target's search sector.

X - A real variable which assumes a random value from a uniform (0,1) probability distribution.

DAMAGE.NUM - An integer variable which is assigned the value 5 if the firer scores a catastrophic kill and the value 6 otherwise.

Coding and Brief Explanation

The coding for event IMPACT is shown below, followed
by a brief explanation:

```
1   UPON IMPACT(A,ID)

2   DEFINE STOPCOUNT AS AN INTEGER VARIABLE

3       DEFINE A AND ID AS INTEGER VARIABLES

4   DEFINE DAMAGE.NUM AND ANSWER AS INTEGER VARIABLES

5   DEFINE WHOCALLED AS AN INTEGER VARIABLE

6   DEFINE DT AS A REAL VARIABLE

7       DEFINE X AS A REAL VARIABLE

8   LET FIP(A)=0

9   CALL DECREMENT.AMMO(A,PROJO(A))

10  LET WHOCALLED=10

11  LET STOPCOUNT=2

12  IF DEFNUM(ID) EQ 1 LET HIT.STATE(ID) = "HIDE" GO TO
MISS ELSE

13  CALL LOC(A)   CALL LOC(ID)

14  IF (WPN.TYPE(A) EQ 6 OR (WPN.TYPE(A) EQ 3 AND PROJO
(A) EQ 1) OR

15      (WPN.TYPE(A) EQ 8 AND PROJO(A) EQ 1) OR WPN.
TYPE(A) EQ 4)
```

16 AND ALIVE.DEAD(A) EQ 1 GO TO MISS2 ELSE

17 CALL LOS GIVEN X.CURRENT(A),Y.CURRENT(A),Z.CURRENT

(A), HTO(A), X.CURRENT(ID),

18 Y.CURRENT(ID), Z.CURRENT(ID), HTTGT(ID), P.V(*)

19 LET PCT.VIS=P.V(1)

20 IF PCT.VIS LT CRITICAL.VALUE

21 GO TO MISS ELSE

22 CALL COMPUTE(A,ID,PCT.VIS)

23   LET FOE(A)=0

24 IF COLOR(A) EQ RED AND X GE 1-PH(A)/2 OR COLOR(A)

EQ BLUE AND X GE 1-PH(A)

25   LET FOE(A)=0

26   LET ALIVE.DEAD(POINTER(ID))=1

27 LET HIT.STATE(ID)="DEAD"

28 LET DAMAGE.NUM=5

29 CALL TALLY.HIT.STATE(ID,DAMAGE.NUM)

30 CALL LIST.UPDATE(0,A,ID,WHOCALLED)

31 IF WPN.TYPE(A) EQ 8 OR WPN.TYPE(A) EQ 3 OR WPN.

TYPE(A) EQ 4

32 CALL STOP.TO.FIRE(A,STOPCOUNT)

33 ALWAYS

34 CALL WE.HIT(A)

```
35              GO TO OUT

36  ELSE

37  'MISS'

38      ''WE DID NOT KILL THE TANK''

39  LET DT=MMM

40  CALL SECTOR.CHECK(ID,A) YIELDING ANSWER

41  IF ANSWER EQ 1  LET DT=NNN  ALWAYS

42  SCHEDULE A DETECT(ID,A) IN DT UNITS

43  IF WPN.TYPE(A) EQ 8 OR WPN.TYPE(A) EQ 3 OR WPN.
TYPE(A) EQ 4

44  CALL STOP.TO.FIRE(A,STOPCOUNT)

45  ALWAYS

46  CALL WE.MISS(A)

47  'MISS2'

48  LET DAMAGE.NUM=6

49  CALL TALLY.HIT.STATE(ID,DAMAGE.NUM)

50  'OUT'

51  CALL DIST(X.CURRENT(A),Y.CURRENT(A),X.CURRENT(ID),
Y.CURRENT(ID)) YIELDING R

52  PRINT 1 LINE WITH TTT,NAME(A),QQ(WPN.TYPE(A)),PROJO
(A),NAME(ID),QQ(WPN.TYPE(ID))
```

95

```
53    ,TIME.V,HIT.STATE(ID), R,X.CURRENT(A),Y.CURRENT(A),

X.CURRENT(ID),Y.CURRENT(ID)       THUS

54
** ** **** * ** **** *** **** **** **** **** **** ****

55   LET TTT = TTT = 1

56   'NEXT'

57   IF HIT.STATE(ID) EQ "DEAD" JUMP AHEAD ELSE

58   LET HIT.STATE(ID) = "     "

59   HERE

60   IF N.READ.ALIVE LE INT.F(BC.COUNT*(1-R.PCT.ATT))

61   OR N.BLUE.ALIVE LE INT.F(RC.COUNT*(1-B.PCT.ATT))

62   SCHEDULE A STOP.SIMULATION IN 15.0 UNITS

63        ALWAYS

64   RETURN     END
```

Lines 1-7 define the event and local variables.

Line 8 resets the firing in progress attribute to 0.

Line 9 reduces the firer's ammunition count by 1 unit
of the type fired.

Lines 10-11 assign values to WHOCALLED and STOPCOUNT.

Line 12 checks the defilade status of the target. If
the target is in full defilade, control is transferred to
the MISS section of the event.

Line 13 updates the X, Y, and Z coordinates of the firer and target.

Lines 14-16 check the ALIVE.DEAD status of missile firers. If the firer is dead prior to the impact of his missile, the missile is considered ineffective and control transfers to the MISS2 section of the event.

Lines 17-19 calculate the percentage of the target visible to the firer.

Lines 20-21 transfer control to the MISS section of the event if the percentage of the target visible to the firer is less than the user defined value.

Line 22 calculates the probability of hit using routine COMPUTE. Note that a hit is synonomous with a kill in the current version of STAR.

Line 23 draws a random number from a Uniform(0,1) probability distribution.

Line 24 tests the random number X against the probability of hit computed in line 22. If X is greater than 1-probability of hit, the target is considered killed and the next block of code is executed; otherwise, control transfers to the MISS section of the event.

Lines 25-28 reset the firer's FOE attribute to 0, change

the ALIVE.DEAD attribute to "DEAD", and assign the value 5

(indicating a catastrophic kill) to the variable DAMAGE.NUM.

Line 29 updates the target's hit counters.

Line 30 removes the target from the firer's detected

list.

Lines 31-32 restart firers who had stopped to fire.

Lines 33-35 determine the next action for the firer and

transfer control to the output section of the event.

Lines 36-38 define and label the MISS section of the

event.

Line 39 sets the detection time of the firer by the

target to a default value of MMM units (currently 6 seconds).

Line 40 determines if the firer is in the target's sector.

Lines 41-42 reset DT to a value of NNN units (currently

2 seconds) if the firer is in the target's sector. A

detection of the firer by the target is scheduled to occur

in DT units.

Lines 43-44 restart firers who had stopped to fire.

Lines 45-49 determine the next action for the firer and

update the target's hit counters.

Lines 50-54 calculate the current range to the target and print information pertinent to the engagement.

Line 55 increments by one the total rounds fired counter.

Lines 56-59 reset the HIT.STATE attribute to four blank spaces if previously set to anything other than "DEAD".

Lines 60-64 stop the simulation if either RED or BLUE forces have reached 100 percent attrition. Control is then returned to the system timer.

I. EVENT HIDE

Description

Event HIDE is used to change the defilade number of a weapon system. Although defilade numbers may be changed anywhere in the program, the use of an event allows the user to change from one condition to another in a specified or random amount of time. Currently, event HIDE is used by events FIRE and IMPACT to bring the element to firing defilade upon engagement of a target, to allow the element to achieve full cover after a firing sequence (dependent on weapon type) and to allow the element to return to turret defilade after an appropriate passage of time under full cover.

99

Currently, only a DEFNUM equal to 1 (fully covered) has
an effect on the execution of STAR. In event STEP.TIME, a
DEFNUM of 1 causes a loss of knowledge of previously
detected targets. In event DETECT two cases exist. If the
observer is in full defilade, the detection is allowed to
occur, but no target selection event is scheduled. This
models the "corporate memory" of the TANK crew when the
TANK returns to turret defilade. If the target is in full
defilade, the target is removed from the observer's list
of detected targets. Event TARGET.SELECT is cancelled if
the selecting TANK is in full defilade. If a potential
target is in full defilade, that target cannot be selected.
Event FIRE is cancelled if the firer is in full defilade. If
the target is in full defilade prior to the impact of the
round, the round will be assessed as a miss in event IMPACT.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value of
the TANK for which a change in defilade is required.

WHOCALLED - An integer variable which assumes values 1
through 5 depending on the type of defilade change required.

WHOCALLED keys the computed go to statement to access the correct label in the code.

Coding and Brief Explanation

The coding for event HIDE is shown below, followed by a brief explanation:

1 UPON HIDE(A,WHOCALLED)

2 DEFINE A AND WHOCALLED AS INTEGER VARIABLES

3 GO TO FULL, TURRET, FIRING, STOP, OTHER PER WHOCALLED

4 'FULL' LET DEFNUM(A) = 1 SCHEDULE A HIDE(A,2) IN
DEF.TIME UNITS RETURN

5 'TURRET' LET DEFNUM(A) = 2 RETURN

6 'FIRING' LET DEFNUM(A) = 3 RETURN

7 'STOP' LET DEFNUM(A) = 4 RETURN

8 'OTHER' LET DEFNUM(A) = 5 RETURN   END

Lines 1-2 define the event and local variables.

Line 3 uses a computed go to statement to access the correct defilade number change instructions in the code.

Line 4 places the TANK in full defilade and schedules a HIDE with a WHOCALLED value of 2 in 20 seconds. When this scheduled HIDE occurs, the WHOCALLED value of 2 will access the next line of code and change the defilade number to 2, turret defilade.

101

Line 5 changes the defilade number to 2, turret defilade.

Line 6 changes the defilade number to 3, firing defilade.

Line 7 changes the defilade number to 4. This type of defilade indicates that the TANK is attempting to acquire concealment.

Line 8 changes the defilade number to 5, a fully exposed position. In all cases above, control is returned to the system timer.

J. EVENT ATTRITION.CHECK

Description

Event ATTRITION.CHECK is used to test the number of BLUE and RED TANKs remaining at specified times during the simulated battle. If either number of survivors falls below the user defined attrition percentage, event STOP. SIMULATION is scheduled in 15 seconds following execution of event ATTRITION.CHECK. This time delay is intended to allow any rounds in the air (especially missiles) to impact prior to halting the simulation.

The user may initially schedule this event in routine MAIN at a point in the battle where he feels that casualties will begin to approach his input attrition thresholds. Alternatively, this event could be scheduled as soon as

102

simulation begins. After the first occurance of the event,

ATTRITION.CHECK schedules itself every DELTA.T seconds.

Local Variables

There are no local variables in this event.

Coding and Brief Explanation

The coding for event ATTRITION.CHECK is shown below,

followed by a brief explanation:

```
1  UPON ATTRITION.CHECK

2  IF N.RED.ALIVE LE INT.F(BC.COUNT*(1-R.PCT.ATT))

3  OR N.BLUE.ALIVE LE INT.F(RC.COUNT*(1-B.PCT.ATT))

4  SCHEDULE A STOP.SIMULATION IN 15.0 UNITS

5  ALWAYS

6  SCHEDULE AN ATTRITION.CHECK IN DELTA.T UNITS

7  RETURN END
```

Line 1 defines the event.

Line 2 compares the number of TANKS remaining in the

RED.ALIVE set to the integer number of TANKs corresponding

to the user's RED attrition percentage input.

Line 3 performs the same test for BLUE TANKs.

Line 4 schedules a halt to the simulation in 15 seconds

if either line 2 or line 3 is true.

Lines 5-7 reschedule this event in DELTA.T seconds and
return control to the system timer.

K. EVENT FINAL.DEATH

<u>Description</u>

Event FINAL.DEATH changes the ALIVE.DEAD attribute of a
TANK to 1 sixty seconds after the TANK sustains a mobility
and firepower kill. This event is scheduled by routine
COMPUTE.

<u>Local Variable</u>

The following is a local variable of this routine:

A - An integer variable containing the pointer value of
the TANK for which an ALIVE.DEAD attribute change is required.

<u>Coding and Brief Explanation</u>

The coding for event FINAL.DEATH is shown below. This
event is self explanatory.

1 UPON FINAL.DEATH(A) DEFINE A AS AN INTEGER VARIABLE
LET ALIVE.DEAD(A)=1 RETURN END

L. EVENT STOP.SIMULATION

<u>Description</u>

Event STOP.SIMULATION is used to return control of the
program to routine MAIN and bring execution of the program

to a halt. This event is scheduled by either event STEP.TIME

or event IMPACT. If event STEP.TIME causes execution of

this event, then a RED TANK has crossed an X and Y coordinate

threshold defined in the event STEP.TIME. If event IMPACT

causes execution of this event, then either the BLUE or RED

forces have reached an attrition point threshold defined in

routine MAIN. In addition to printing the elapsed simulated

time in seconds, the number of line of sight calls are

printed and the current attrition values of each weapon

system are printed.

The coding for event STOP.SIMULATION is shown below. There

are no local variables in this event and the code is self

explanatory.

```
1   UPON STOP.SIMULATION

2   PRINT 1 LINE WITH TIME.V AND LIN THUS
SIMULATION STOPPED AT  *****.****   LOS CALLS = ****************

3   SKIP 3 OUTPUT LINES

4   FOR EACH TANK IN TANKS CALL LOC(TANK)

5   LIST ATTRIBUTES OF EACH TANK IN TANKS

6   STOP

7   END
```

M. ROUTINE SET.SECTOR

Description

Routine SET.SECTOR is used to provide unit vector endpoints in the direction of the left and right search sector limits of a particular weapon system. A simple rotation matrix is applied to the primary direction of search. This matrix multiplication produces the required endpoints. Routine SET.SECTOR is used when weapons systems are created and any time that the primary direction of search changes. Currently, the MAIN routine and routine CHG.SEC.SEARCH use routine SET.SECTOR for these purposes. See event STEP.TIME for conditions that activate routine CHG.SEC.SEARCH.

Local Variables

The following are local variables of this routine:

TANK - An integer variable containing the pointer value of the weapon system for which the left and right sector limits are being developed.

X - The X coordinate of a unit vector in the direction of the current primary direction of search.

Y - The Y coordinate of a unit vector in the direction of the current primary direction of search.

106

A - A real variable which has the value of the cosine of one-half the current search sector angle.

B - A real variable which has the value of the sine of one-half the current search sector angle.

WIDTH - One-half the current search sector angle in radians.

Coding and Brief Explanation

The coding for routine SET.SECTOR is shown below, followed by a brief explanation.

```
1  ROUTINE SET.SECTOR(TANK)

2  DEFINE TANK AS AN INTEGER VARIABLE

3  DEFINE X,Y,WIDTH,A,B AS REAL VARIABLES

4  LET X = COS.F(PRI.DIR(TANK))  LET Y = SIN.F(PRI.DIR
(TANK))

5  LET WIDTH =(AREA/2.0) * PI.C * 2.0 / 360.0

6  LET A = COS.F(WIDTH)  LET B = SIN.F(WIDTH)

7  LET X.R(TANK) = A*X + B*Y

8  LET Y.R(TANK) = -B * X + A*Y

9  LET X.L(TANK) = A*X + (-B) * Y

10  LET Y.L(TANK) = B*X + A *Y

11  RETURN END
```

Lines 1-3 define the routine and local variables.

Line 4 calculates the endpoints of a unit vector in the direction of the primary direction of search.

Line 5 calculates the value of one-half the current search sector angle.

Line 6 calculates the sine and cosine required for the rotation matrix.

Lines 7-11 calculates the X and Y endpoints of unit vectors in the direction of the left and right search sector limits, respectively. Control is then returned to the routine MAIN or routine CHG.SEC.SEARCH. The rotation matrix, R, represented by these lines of code is shown below. Note that any positive or negative angle may be substituted for WIDTH.

$$R = \begin{bmatrix} COS(WIDTH) & SIN(WIDTH) \\ -SIN(WIDTH) & COS(WIDTH) \end{bmatrix}$$

N. ROUTINE LOC

Description

Routine LOC is a simplified movement routine used in the basic version of STAR. It is called by numerous routines and events throughout STAR, and is used to update

the X, Y, and Z coordinates of a particular TANK. In the
current version of STAR, BLUE TANKs do not move. An
assumption of instantaneous acceleration to constant speed
is made, as is the assumption of instantaneous deceleration to
a speed of zero. The Z coordinate, elevation, is obtained
from FORTRAN routine ELEV which resides on disk. An
elevation floor of 250 meters is used to save line of sight
computation time in FORTRAN routine LOS. The movement
routine for production versions of STAR will be a FORTRAN
routine.

Local Variables

The following is a local variable of this routine:

TANK - An integer variable containing the pointer value
of the element for which a location update is required.

Coding and Brief Explanation

The coding for routine LOC is shown below, followed by a
brief explanation:

    1   ROUTINE LOC(TANK)

    2   DEFINE TANK AS AN INTEGER VARIABLE

    3   IF ALIVE.DEAD(TANK) EQ 1 OR COLOR(TANK) EQ BLUE RETURN
ELSE

```
4  LET X.CURRENT(TANK)=X.CURRENT(TANK)+(TIME.V-T.SPD

(TANK))*(SPD(TANK)

5     *COS.F(DIR.OF.MVMT(TANK)))

6  LET Y.CURRENT(TANK)=Y.CURRENT(TANK)+(TIME.V-T.SPD

(TANK))*(SPD(TANK)

7     *SIN.F(DIR.OF.MVMT(TANK)))

8  LET T.SPD(TANK)=TIME.V

9  CALL ELEV GIVEN X.CURRENT(TANK),Y.CURRENT(TANK),

ZH(*)

10  IF ZH(1) LE 250.0 LET ZH(1) = 250.0 ALWAYS LET

Z.CURRENT(TANK) = ZH(1)

11  RETURN

12  END
```

Lines 1-2 define the routine and local variable.

Line 3 insures that the tank is alive and is not BLUE
before proceeding.

Lines 4-5 calculate the new X coordinate by adding to
the current X coordinate the total distance traveled since
the speed was last set.

Lines 6-7 calculate the current Y coordinate in a similar
manner.

Lines 10-11 calculate the current elevation using

FORTRAN routine ELEV. A minimum elevation of 250 meters is

set and control is returned to the calling routine or event.

O.   ROUTINE DIST

Description

Routine DIST is used by many events and routines in

STAR when the range in meters between two selected TANKs

is required.

Local Variables

The following are local variables of this routine:

X1 - The current X coordinate of the first TANK.

Y1 - The current Y coordinate of the first TANK.

X2 - The current X coordinate of the second TANK.

Y2 - The current Y coordinate of the second TANK.

DISTANCE - The return argument of routine DIST which

represents the range in meters between the two selected

TANKs.

Coding and Brief Explanation

The coding for routine DIST is shown below. This

routine is self explanatory.

```
1  ROUTINE FOR DIST GIVEN X1,Y1,X2,Y2 YIELDING DISTANCE

2      LET DISTANCE=SQRT.F((X1-X2)**2 + (Y1-Y2)**2)

3  RETURN

4  END
```

P. ROUTINE SECTOR.CHECK

Description

Routine SECTOR.CHECK uses the definition of the dot

product of two vectors to determine if a target is within

an observer's search sector. In the current version of

STAR, a search sector is defined in routing MAIN to be 90

degrees, the cosine of which is 0. Recall that the cosine of

an angle can be defined as:

$$\text{COSINE } \theta = (A \cdot B) / |AB|$$

where θ is the angle between two vectors, A and B, A·B is

the dot product of the two vectors, and |AB| is the magnitude

of a vector constructed between the end points of the given

vectors. If the angle between the observer's left sector

limit and the target has a cosine of less than 0, or if the

angle between the observer's right sector limit and target

has a cosine of less than 0, then the target is not within

the observer's sector. Since routine CARDIO considers only

those targets within sector, routine SECTOR.CHECK is used as

112

a filter prior to calling routine CARDIO in event STEP.TIME.

Routine SECTOR.CHECK is currently used by events STEP.TIME

and IMPACT.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value of

the observer TANK.

B - An integer variable containing the pointer value

of the target TANK.

ANSWER - An integer variable which assumes the value 1 if

the target is within the observer's sector, and the value 0

if the target is not in sector.

X.T - The X coordinate of the endpoint of the vector

from the observer to the target.

Y.T - The Y coordinate of the endpoint of a vector from

the observer to the target.

C.LEFT - The cosine of the angle between the target and

the observer's left sector limit.

C.RIGHT - The cosine of the angle between the target

and the observer's right sector limit.

R - The magnitude, $|AB|$, described above.

Coding and Brief Explanation

The coding for routine SECTOR.CHECK is shown below, followed by a brief explanation:

```
1  ROUTINE SECTOR.CHECK(A,B) YIELDING ANSWER

2  DEFINE A,B, AND ANSWER AS INTEGER VARIABLES

3  DEFINE X.T, Y.T, C.LEFT, C.RIGHT, AND R AS REAL
VARIABLES

4  LET ANSWER = 1

5  LET X.T = X.CURRENT(B) - X.CURRENT(A)    LET Y.T =
Y.CURRENT(B) - Y.CURRENT(A)

6  LET R= SQRT.F(((ABS.F(X.L(A)+X.T))**2.0)+((ABS.F
(Y.L(A)+Y.T))**2.0))

7  LET C.LEFT= (X.L(A)*X.T+Y.L(A)*Y.T)/R

8  LET C.RIGHT= (X.R(A)*X.T+Y.R(A)*Y.T)/R

9  IF C.LEFT LT CONSTANT OR C.RIGHT LT CONSTANT LET
ANSWER=0 ALWAYS RETURN END
```

Lines 1-3 define the routine and local variables.

Line 4 sets the default value of ANSWER to 1.

Line 5 calculates the endpoints of a vector to the target with respect to the observer.

Line 6 calculates the magnitude, $|AB|$, described above.

114

Lines 7-8 calculate C.LEFT and C.RIGHT respectively.

Line 9 tests C.LEFT and C.RIGHT against CONSTANT, the cosine of 90 degrees in the current version. If either cosine is less than zero, the target is considered to be out of sector, ANSWER is set to 0, and control is returned to the calling event.

Q. ROUTINE CHG.SEC.SEARCH

Description

Routine CHG.SEC.SEARCH is called by event STEP.TIME whenever a particular vehicle has no currently detected targets in its target list. CHG.SEC.SEARCH changes a vehicle's primary direction of search to allow a vehicle to search areas other than the original area assigned to the vehicle.

The primary direction of search is in relation to the vehicle's direction of movement. The attribute PRI.DIR is the vehicle's primary direction of search. The attribute DIR.OR.MVMT is the vehicle's direction of movement. The routine CHG.SEC.SEARCH changes the PRI.DIR attribute of a specified vehicle from zero to $\pi/4$, $\pi/4$ to $-\pi/4$, or $-\pi/4$

115

to zero depending on the current PRI.DIR attribute of the

vehicle when the routine is called.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value of

the vehicle for which a change of sector is to be

accomplished.

XYZ - An integer variable used to direct a computed go

to statement to the proper label.

The coding for CHG.SEC.SEARCH is shown below, followed

by a brief explanation:

```
1  ROUTINE CHG.SEC.SEARCH(A)

2  DEFINE A AND XYZ AS INTEGER VARIABLES

3  LET XYZ = SIGN.F(PRI.DIR(A)-DIR.OF.MVMT(A))

4  LET XYZ=XYZ+2

5  GO TO C1,C2,C3  PER XYZ

6  'C1' LET PRI.DIR(A) = 0. + DIR.OF.MVMT(A)  GO TO SET

7  'C2' LET PRI.DIR(A) = PI.C/4 + DIR.OF.MVMT(A)  GO
TO SET

8  'C3' LET PRI.DIR(A) = -PI.C/4. + DIR.OF.MVMT(A)  GO
TO SET
```

```
9   'SET'     CALL SET.SECTOR(A)

10  RETURN    END
```

Lines 1-2 define the routine and local variables.

Line 3 uses the SIGN.F system function to set the initial

value of XYZ.   If the argument is zero XYZ=0, if the argument

is positive XYZ=1, if the argument is negative XYZ=-1.

Line 4 adds 2 to XYZ for use in the computed go to

statement

Line 5 directs the routine to label 'C1' if XYZ=1, to

label 'C2' if XYZ=2, and to label 'C3' if XYZ=3.

Lines 6-8 set the new primary direction of search on the

appropriate vehicle.

Line 9 calls routine SET.SECTOR which resets the sector

limits of the appropriate vehicle based on the new primary

direction of search.

Line 10 returns control to the calling routine or event.

R.   ROUTINE CARDIO

Description

Routine CARDIO is called by event STEP.TIME whenever it

is necessary to determine how long it will take (in seconds)

for an observer to detect a given target.   This routine uses

117

the assumption that detection is a random phenomenom.   The

basic detection equation used in the routine is:

$$P(d)=1-exp(-LAMBDA * t)$$

where: P(d) is the probability of detection

LAMBDA is the detection rate

t is the time to detect

The routine draws a random number from a uniform(0,1)

distribution as the value of P(d), then solves for t and

returns the value of t to STEP.TIME.

Routine CARDIO assigns a uniform probability of an

observer looking in each 30 degree portion of his search

area.   (Area size in degrees is user specified.)   Thus,

if the observer's search area is 90 degrees he has a one-

third probability of looking in each 30 degree portion of

his search area at a specified time.   If the user specifies

360 degrees as the search area, routine CARDIO utilizes a

cardioid distribution to determine probability of looking in

any particular 30 degree portion of the search area.

The value of LAMBDA used in the detection equation is

calculated using the following formula $\lceil$ Ref. 10 $\rceil$:

$$LAMBDA = .003 + \frac{1.088}{1.453 + TC.FACTOR*(.5978 + 2.138*(RR**2)- .5038*X.VELOCITY)}$$

where: T.C.FACTOR is the complexity of the terrain between

   observer and target

RR is the apparent range to the target in kilometers

X.VELOCITY is the crossing velocity of the target

If the target is within 300 meters of the observer,

detection is assumed to be essentially immediate and a time

of one second is returned to event STEP.TIME. If the target

is beyond 4000 meters of the observer, detection is assumed

to be essentially infinite and a time of 99 is returned to

event STEP.TIME.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value

of the observer.

ANGLE - A real variable containing the angle in radians

between the observer's primary direction of search and the

target.

AT - A real variable equivalent to $1/2\,\pi$.

DD - A real variable used as a correction factor to

calculate the apparent range to the target.

DET.TIME - A real variable containing the time to

detect a given target.

LAMBDA - A real variable containing the value of the detection rate.

AREA - A real variable containing the observer's search area in degrees.

B - An integer variable containing the pointer value of the target vehicle.

BT - A real variable containing the value $3/8\pi$.

DENOM - A real variable containing the denominator of the detection rate equation.

P.SUB.K —A real variable containing a value representing the probability an observer is looking in a 30 degree portion of his search sector.

PER.FULL.EXPO - Percent of the target that is exposed (value returned from routine PERCENT).

RR - A real variable representing the angle from the observer to the target.

TGT.ELEMENT - A real variable representing the angle from the observer to the target.

X.VELOCITY - A real variable containing the crossing velocity of the target vehicle.

MT - A real variable containing the value $6/\pi$.

PCT.VIS - A real variable valued between 0 and 1 which represents the percentage of the target which is visible to the observer.

R - A real variable containing the range to the target in meters passed into routine CARDIO from event STEP.TIME.

T.C.FACTOR - A real variable representing the complexity of the terrain between the observer and target. Value is obtained by routine CARDIO by calling routine TER.COMPLEXITY.

X - A real variable containing the value of a uniform random variable between 0 and 1 passed into routine CARDIO from event STEP.TIME.

ZL - A real variable containing the constant .000001 used to prevent division by zero in the routine.

The coding for CARDIO is show below, followed by a brief explanation:

```
1   ROUTINE CARDIO(A,B,R,PCT.VIS,X)  YIELDING DET.TIME

2   DEFINE R.PCT.VIS,AND X AS REAL VARIABLES

3   DEFINE AREA AS A REAL VARIABLE

4   DEFINE A AND B AS INTEGER VARIABLES

5   IF COLOR(A) EQ RED

6       LET AREA=R.AREA

7   ALWAYS
```

```
8   IF COLOR(A) EQ BLUE

9       LET AREA=B.AREA

10  ALWAYS

11  IF R LE 300.0 LET DET.TIME=1.0

12  RETURN

13  ELSE

14  IF R GT 4000.0  LET DET.TIME=99.0

15  RETURN

16  ELSE

17  LET ZL=.000001

18  LET DD = PCT.VIS

19  IF DD LE 0. LET DD=ZL  ALWAYS LET RR=(R/DD)/1000.

20  LET TGT.ELEMENT=ARCTAN.F ((Y.CURRENT(B)-Y.CURRENT

(A)),(X.CURRENT(B)-

21             X.CURRENT(A)))

22  IF AREA EQ 360.

23      GO TO FULL.CARDIOID

24  ELSE

25  LET P.SUB.K =  30./AREA

26  GO TO CONTINUE

27  'FULL.CARDIOID'

28  LET ANGLE=ABS.F(TGT.ELEMENT-PRI.DIR(A))
```

122

```
29  LET MT=6./PI.C

30  LET BT=3./(8.*PI.C)

31  LET AT=1./(2.*PI.C)

32  LET P.SUB.K=(BT/MT)+AT*(SIN.F(ANGLE+(1/MT))-SIN.F
(ANGLE))

33      IF P.SUB.K LT 0.

34      LET P.SUB.K=0

35  ALWAYS

36  'CONTINUE'

37  LET X.VELOCITY=ABS.F(SPD(B)*SIN.F(TGT.ELEMENT-DIR.
OF.MVMT(B)))

38  CALL TER.COMPLEXITY(X.CURRENT(A),Y.CURRENT(A),
X.CURRENT(B),Y.CURRENT(B))

39          YIELDING T.C.FACTOR

40  LET DENOM=1.453+T.C.FACTOR*(.5978+2.188*(RR**2)-
.5038*X.VELOCITY)

41  IF DENOM LE ZL

42  LET DENOM=ZL

43  ALWAYS

44  LET LAMBDA= .003+1.088/DENOM

45  IF LAMBDA LE 0.0 LET DET.TIME=0.0 RETURN ELSE

46  IF SPD(A) GT 0.
```

```
47  LET LAMBDA=LAMBDA/2

48  ALWAYS

49  CALL PERCENT(A,B) YIELDING PER.FULL.EXPO

50  LET LAMBDA=LAMBDA*PER.FULL.EXPO*P.SUB.K

51  IF LAMBDA LE 0.

52  LET LAMBDA=ZL

53  ALWAYS

54  LET DET.TIME=LOG.E.F(1.-X)/(-LAMBDA)

55  RETURN

56  END
```

Lines 1-4 define the routine and local variables.

Lines 5-10 assign appropriate search area to the observer based on user input from the MAIN program.

Lines 11-16 check range to the target versus upper and lower threshold values, and returns appropriate DET.TIME if threshold values are exceeded.

Line 17 sets value of ZL to prevent division by zero.

Line 18-19 calculate apparent range to the target.

Line 20 calculates angle between observer and target.

Lines 21-23 check to see if user has specified use of cardiod distribution and if so transfer to 'FULL.CARDIOD' label.

Lines 24-25 set value of P.SUB.K.

Lines 27-35 calculate value cf P.SUB.K using the cardiod distribution.

Line 37 calculates crossing velocity of the target.

Lines 38-39 call routine TER.COMPLEXITY which returns with the value of T.C.FACTOR.

Lines 40-43 calculates the value of the denominator of the detection rate equation.

Lines 44-45 calculate the initial value of the detection rate. If the detection rate is less than or equal to 0, a time of 0.0 is returned, indicating immediate detection, to event STEP.TIME.

Lines 46-48 check to see if the observer is moving; if so, the detection rate is halved.

Line 49 calls routine PERCENT which returns with the value of PER.FULL.EXPO.

Lines 50-53 calculate final value of detection rate using P.SUB.K,PER.FULL.EXPO, and initial LAMBDA.

Line 54 calculates detection time.

Line 55 returns control to the calling event.

S. ROUTINE PERCENT

Routine PERCENT is a dummy routine called by routine
CARDIO yielding the real variable PER.FULL.EXPO.

PER.FULL.EXPO is a real variable which indicates the
percentage of the movement trace of the target where the
target is fully exposed (not concealed or covered), and
is an integral part of the detection rate calculation in
routine CARDIO.

Routine PERCENT in its present form always returns a
value of 1.0 for PER.FULL.EXPO. Upon availability of
micro-terrain data, this routine will return a real value
between 0.0 and 1.0 to routine CARDIO by accessing the
micro-terrain data.

Local Variables

The following are local variables of this routine:

PER.FULL.EXPO - A real variable indicating the percentage
of the target's movement trace that the target is fully
exposed.

X - An integer variable containing the pointer value of
the observer.

Y - An integer variable containing the pointer value of
the target.

The coding for PERCENT is shown below, followed by a brief explanation:

```
1  ROUTINE FOR PERCENT GIVEN X AND Y YIELDING PER.FULL.EXPO
2  LET PER.FULL.EXPO=1
3  RETURN
4  END
```

Line 1 defines the routine and variables passed from routine CARDIO.

Line 2 defines the value of PER.FULL.EXPO to be 1.0.

Lines 3-4 return control to the calling routine or event.

T. ROUTINE TER.COMPLEXITY

Description

Routine TER.COMPLEXITY is a dummy routine called by routine CARDIO yielding the real variable T.C.FACTOR.

T.C.FACTOR is a real variable which indicates the complexity of the terrain between the observer and the target, and is an integral part of the detection rate calculations in routine CARDIO.

Routine TER.COMPLEXITY in its present form always returns a value of 1.0 for T.C.FACTOR. Upon availability

127

of micro-terrain data this routine will return a value

between 1.0 and 7.0 to routine CARDIO by accessing the

micro-terrain data.

Local Variables

The following are local variables of this routine:

T.C.FACTOR - A real variable indicating the complexity

of terrain between observer and target.

X - A real variable passed into TER.COMPLEXITY from

routine CARDIO containing the current X coordinate of the

observer.

Y - A real variable passed into TER.COMPLEXITY from

routine CARDIO containing the current Y coordinate of the

observer.

Z - Z real variable passed into TER.COMPLEXITY from

routine CARDIO containing the current X coordinate of the

target.

W - A real variable passed into TER.COMPLEXITY from

routine CARDIO containing the current Y coordinate of the

target.

The coding for TER.COMPLEXITY is shown below, followed

by a brief explanation:

```
1  ROUTINE FOR TER.COMPLEXITY GIVEN X,Y,Z,W YIELDING
T.C.FACTOR

   2  LET T.C.FACTOR=1.

   3  RETURN

   4  END
```

Line 1 defines the routine and variables passed into it
from routine CARDIO.

Line 2 defines the value of T.C.FACTOR to be 1.0.

Lines 3-4 return control to the calling routine.

U.   ROUTINE LIST.UPDATE

Description

Routine LIST.UPDATE is the means by which the list of
detected targets for each weapon system in the simulation
is kept current. The calling arguments are used to control
the actions within the routine.

Routine LIST.UPDATE serves three distinct functions:
addition of a target to a specified weapon system's target
list, removal of a target from a specified weapon system's
list and scheduling of event TARGET.SELECT.

Removal of a target from a list occurs when one of two
conditions exist: either line of sight to the target no

longer exists or the target is considered "dead". If the
target in question was not previously placed in the weapon
system's target list, no action is taken in regard to
updating the weapon system target list.

Addition of a target to a list occurs when the target
is detected and line of sight exists to the target.

Either removal or addition of a target to a list may
result in the scheduling of event TARGET.SELECT based on
the value of the argument WHOCALLED passed into routine
LIST.UPDATE by the calling event or routine a WHOCALLED
value less than or equal to 1 will cause the routine to
perform its function without event TARGET.SELECT being
scheduled. A WHOCALLED argument of 2 or greater will cause
event TARGET.SELECT to be scheduled prior to return to the
calling event or routine.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value
of the weapon system whose list is to be updated.

B - An integer variable containing the pointer value of
the target weapon system to be added or removed from the
list.

COUNT - An integer variable set to equal the location of a particular target in a target list.

I - An integer variable used as a DO LOOP counter.

SIZE - An integer variable which indicates the dimension of a target list.

FLAG - An integer variable set to 1 if line of sight does not exist to a particular target or if the target in question is "dead", 0 otherwise.

LOSE - An integer variable set to 1 if line of sight exists to a target and to 0 if line of sight does not exist. The value of LOSE is passed to routine LIST.UPDATE by the calling event or routine.

WHOCALLED - An integer variable indicating the requirement for routine LIST.UPDATE to schedule event TARGET.SELECT.

The coding for LIST.UPDATE is shown below, followed by a brief explanation:

```
 1   ROUTINE LIST.UPDATE(LOSE,A,B,WHOCALLED)
 2   DEFINE I AS AN INTEGER VARIABLE
 3   DEFINE A,B,LOSE,SIZE,COUNT,FLAG,AND WHOCALLED AS
INTEGER VARIABLES
 4   LET LIST(*) = TARGET (NAME(A),1)   ' '   ACCESS
CORRECT LIST ' '
```

```
5  IF LOSE EQ 0 OR ALIVE.DEAD(B) EQ 1

6        LET FLAG=1

7  REGARDLESS

8        LET SIZE = DIM.F(LIST(*))

9  FOR I = 1 TO SIZE, DO

10        IF B EQ LIST(I)

11              LET COUNT = I

12              GO TO OUT.OF.LOOP

13        ELSE

14        LOOP

15 IF FLAG EQ 1

16 LET LIST(*) = 0

17        RETURN

18 ELSE

19      IF LIST(1) EQ 0

20              LET LIST(1) = B

21              LET SIZE = 1

22 GO TO ALWAYS

23        ELSE

24              FOR I = 1 TO SIZE    LET TEMP.TGT(*) =
LIST(I)

25              LET TEMP.TGT(SIZE+ 1) = B
```

132

```
26  RELEASE LIST(*)

27      RESERVE LIST(*) AS SIZE + 1

28      LET TARGET(NAME(A),1) = LIST(*)

29      FOR I = 1 TO SIZE + 1  LET LIST(I) = TEMP.TGT(1)

30  'ALWAYS'

31      LET LIST(*) = 0

32  IF WHOCALLED LE 1 RETURN

33  ELSE

34  SCHEDULE A TARGET.SELECT(A) IN MIN.F(UNIFORM.F

(0.,SQRT.F(SIZE),2),

35  SQRT.F(5.0)) UNITS

36      RETURN

37  'OUT.OF.LOOP'

38  IF FLAG EQ 1

39      IF SIZE EQ 1 LET LIST(1) = 0

40              LET LIST(*) = 0

41      RETURN

42  ELSE

43      FOR I = 1 TC SIZE   LET TEMP.TGT(I) = LIST(I)

44  RELEASE LIST(*)

45      RESERVE LIST(*) AS SIZE - 1

46      LET TARGET (NAME(A),1) = LIST(*)
```

```
47        FOR I = 1 TO SIZE = 1, DO

48            IF I LT COUNT

49                LET LIST(I) = TEMP.TGT(I)

50  GO TO NEXT

51            ELSE

52                LET LIST(I) = TEMP.TGT(I+1)

53  'NEXT'

54            LOOP

55  REGARDLESS

56        LET LIST(*) = O

57  IF WHOCALLED   LE 1 OR FLAG EQ 1 RETURN ELSE

58  SCHEDULE A TARGET.SELECT(A) IN MIN.F(UNIFORM.F
(O.,SQRT.F(SIZE),2),

59  SQRT.F(5.0)) UNITS

60        RETURN

61        END
```

Lines 1-3 define the routine and local variables.

Lines 4 accesses the appropriate target list.

Lines 5-7 set FLAG to 1 if line of sight does not exist

or if the target in question is"dead".

Lines 8-18 search the accessed target list. If the target in question is in the list, COUNT is set equal to the target's location in the list and control is transferred to the 'OUT.OF.LOOP' label. If the target is not in the list, and FLAG=1, the list is released and control returned to the calling routine or event.

Lines 19-22 check to see whether the accessed list has any elements in it. If not, and all previous checks have been passed, the target in question is placed in the list and control transfers to the 'ALWAYS' label.

Lines 23-29 replace the accessed list with a new list which contains the target in question, and releases the core storage used by the old list for further use in the program.

Lines 30-36 check the WHOCALLED argument and schedule event TARGET.SELECT if appropriate, control is returned to the calling event or routine.

Lines 37-56 remove the appropriate target from the accessed target list and release the list.

Lines 57-59 check the WHOCALLED argument and schedule event TARGET.SELECT if appropriate.

Lines 60-61 return control to the calling event or routine.

V.   ROUTINE PROXIMITY.DETECT

Description

Routine PROXIMITY.DETECT is called by event DETECT
whenever a normal detection of a TANK occurs. This routine
simulates the phenomenon of close vicinity detection. The
rationale behind this routine is the fact that detection
of an element on the battlefield causes detection of other
elements within some relatively close, specified range. A
single target selection is scheduled as a result of detect-
ing a cluster of targets.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value
of the TANK effecting a near vicinity detection.

B - An integer variable containing the pointer value
of the detected threat. Other elements close to this
threat may be detected, if they meet certain close vicinity
criteria (currently a square 150 meters on a side centered
at the detected target).

WHOCALLED - An integer variable used to control the
execution of a target selection in routine LIST.UPDATE.

A value of 0 prevents a target selection, while a value of

10 allows a target selection.

X.SAMPLE - A real variable containing the current X

coordinate of the detected threat, B.

Y.SAMPLE - A real variable containing the current Y

coordinate of the detected threat, B.

Coding and Brief Explanation

The coding for routine PROXIMITY.DETECT is shown below,

followed by a brief explanation.

```
1   ROUTINE PROXIMITY.DETECT(A,B)

2   DEFINE A AND B AS INTEGER VARIABLES

3   DEFINE X.SAMPLE AND Y.SAMPLE AS REAL VARIABLES

4   DEFINE WHOCALLED AS AN INTEGER VARIABLE

5   IF ALIVE.DEAD(B) EQ 1 RETURN ELSE

6   LET X.SAMPLE=X.CURRENT(B)

7   LET Y.SAMPLE=Y.CURRENT(B)

8   IF COLOR(A) EQ BLUE

9   FOR EACH TANK IN PLT.UNIT(PLT(B  ))

10                                  WITH ABS.F(X.CURRENT
TANK)-X.SAMPLE) LE 75.0

11      AND ABS.F(Y.CURRENT(TANK)-Y.SAMPLE) LE 75.0, DO
```

```
12  IF TANK EQ B LET WHOCALLED = 10 ALWAYS

13  CALL LIST.UPDATE(1,A,TANK,WHOCALLED)

14  LET WHOCALLED = 0

15     LOOP    JUMP AHEAD

16  ELSE

17  FOR EACH TANK IN PLT.UNIT(PLT(B   ))

18                              WITH ABS.F(X.CURRENT
(TANK)-X.SAMPLE) LE 75.0

19     AND ABS.F(Y.CURRENT(TANK)-Y.SAMPLE) LE 75.0, DO

20  IF TANK EQ B LET WHOCALLED = 10 ALWAYS

21  CALL LIST.UPDATE(1,A,TANK,WHOCALLED)

22  LET WHOCALLED = 0

23  LOOP

24  HERE

25  RETURN

26  END
```

Lines 1-4 define the routine and local variables.

Line 5 insures that the currently detected threat,
B, is still alive prior to proceeding.

Lines 6-7 place the X and Y coordinates of the currently
detected threat in holding variables.

Line 8 accesses the BLUE section of this routine.

Lines 9-11 search B's platoon for TANKs within a 150 meter square centered on B.

Lines 12-14 test if the TANK meeting the distance criteria is in fact B. If so, routine LIST.UPDATE is accessed with a WHOCALLED value of 10. This insures a potential target selection based on the detection of B. If the TANK meeting the distance criteria is merely a member of B's platoon, routine LIST.UPDATE is accessed with a WHOCALLED value of 0 and no target selection is scheduled as a result of close vicinity detections. After searching the platoon, control is returned to event DETECT.

Line 16 accesses the RED portion of the routine.

Lines 17-25 are virtually identical to lines 9-15, but apply to RED weapon systems.

W. ROUTINE T72.TACTICS

Description

Routine T72.TACTICS is the tactical routine used to simulate platoon level fire control within the RED tank force. Routine T72.TACTICS is called by event TARGET.SELECT and returns to event TARGET.SELECT the integer variable ANSWER. If the value of ANSWER=1, the RED tank in question

139

will go thru the normal target selection process. An

ANSWER of 0 indicates to event TARGET.SELECT that the

tank in question has acquired a target designated by his

platoon leader.

Routine T72.TACTICS sequences through several checks

to determine if it is appropriate for the RED tank in

question to have designated as his highest priority target

his platoon leader's currently selected target. All four

of the following conditions must be met for a platoon

member to have his platoon leader's target designated as

his own.

1. The platoon leader must be "alive".

2. Line of sight must exist between platoon leader and

member.

3. The platoon leader must have a target selected.

4. The platoon member must have the proper ammunition

type available.

If any of these four conditions are not met, ANSWER is set

to 1, and control returned to the calling event.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value of the platoon member.

ANSWER - An integer variable assigned a value of 1 if the vehicle is to go through a normal target select, and 0 if the tank has acquired his platoon leader's target.

LOSE - An integer variable indicating the existence of line of sight; 1 indicates line of sight exists, 0 indicates line of sight does not exist.

AIM - An integer variable returned to T72.TACTICS from routine COMMO.PASS.TARGET containing the pointer value of the platoon leader's currently selected target.

RESULT - An integer variable returned to T72.TACTICS by routine AMMO.CHECK indicating the availability of appropriate ammunition. If RESULT equals 0, the proper ammunition is not available; RESULT equals 1 otherwise.

The coding for T72.TACTICS is shown below, followed by a brief explanation:

```
1   ROUTINE T72.TACTICS(A)  YIELDING ANSWER
2   DEFINE A, ANSWER, AND RESULT AS INTEGER VARIABLES
3   DEFINE AIM AS AN INTEGER VARIABLE
4   DEFINE LOSE AS AN INTEGER VARIABLE
```

141

```
 5  LET ANSWER = 0

 6  IF PLTLDR(A) NE A AND COCDR(A) NE A

 7  IF ALIVE.DEAD(PLTLDR(A)) EQ 1

 8  FOR EACH TANK IN PLT.UNIT(PLT(A  ))

 9  LET PLTLDR(TANK) = TANK LET ANSWER = 1 RETURN ELSE

10  CALL LOC(A)  CALL LOC(PLTLDR(A))

11  CALL LOS  GIVEN X.CURRENT(A),Y.CURRENT(A),Z.CURRENT
(PLTLDR(A)),HTTGT(PLTLDR(A)),

12  X.CURRENT(PLTLDR(A)),Y.CURRENT(PLTLDR(A)),Z.CURRENT
(PLTLDR(A)),HTTGT(PLTLDR(A)),

13      P.V(*)

14  LET PCT.VIS=P.V(1)

15  IF PCT.VIS GT CRITICAL.VALUE

16  LET LOSE=1  JUMP AHEAD  ELSE  LET LOSE=0  HERE

17  IF LOSE EQ 0  LET ANSWER= 1  RETURN  ELSE  GO TO COMMO

18  'COMMO'  CALL COMMO.PASS.TGT(A) YIELDING AIM

19  IF AIM NE 0

20  CALL AMMO.CHECK(A,PROJO(PLTLDR(A))) YIELDING RESULT

21  IF RESULT EQ 0 LET ANSWER = 1 RETURN ELSE

22  LET FOE(A) = AIM LET PROJO(A) = PROJO(PLTLDR(A))

23  LET SCHED(A) = 1
```

24 SCHEDULE A FIRE(A,AIM) IN MAX.F(UNIFORM.F(10.,20.,3),
UNIFORM.F(8.,16.,4)) UNITS

25 LET CHECK.TIME(A) = TIME.A(FIRE) RETURN

26 ELSE ALWAYS LET ANSWER=1   RETURN   END

Lines 1-5 define the routine and local variables and set
the initial value of ANSWER.

Line 6 checks to determine if the tank in question is a
platoon leader or a company commander; if so, control is
transferred to line 26, ANSWER is set to 1, and control is
returned to the calling event.

Lines 8-9 check to see if the appropriate platoon leader
is "dead". If so, each member of that particular platoon
is made a "platoon leader". This causes platoon members
whose platoon leader is destroyed to always go through the
normal sequence of selecting a target from their own target
lists.

Lines 10-17 update the location of the platoon leader
and platoon member and determine if line of sight exists
between the two. If line of sight does not exist, ANSWER
is set to 1 and control is returned to event TARGET.SELECT.

Line 18 calls routine COMMO.PASS.TARGET and has returned
the platoon leader's current FOE attribute (AIM).

143

Lines 19-21 determine if the platoon leader has a
current FOE and checks to see if the platoon member has
the appropriate ammunition available. If the proper
ammunition is not available, ANSWER is set to 1, and
control is returned to the calling event.

Lines 22-25 set the proper FOE, PROJO, SCHED, and
CHECK.TIME attributes on the platoon member and schedule
the event FIRE to occur in the maximum of the lay and load
time of the weapon system.

Line 26 returns control to the calling event.

X.   ROUTINE COMMO.PASS.TGT

Description

Routine COMMO.PASS.TGT is used by routine T72.TACTICS
for RED TANK fire control. In the current version of STAR,
only weapons of type T72 use this routine. The selecting
TANK, if not a platoon leader attempts to acquire as his
FOE the FOE of his platoon leader. Line of sight must
exist between the selecting tank and his platoon leader,
the platoon leader must be alive, the potential target must
be within the opening range of the platoon member and the
platoon member must have a FOE. If any of these four
conditions is not met, control is returned to routine

T72.TACTICS with AIM = 0. This directs T72.TACTICS to allow a normal target selection for the firer. If all conditions are met, AIM is set to the pointer value of the platoon leader's FOE and T72.TACTICS allows an attempted engagement of this FOE.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value of the TANK attempting to acquire the FOE of his platoon leader.

BAIM - An integer variable containing the pointer value of the platoon leader's FOE.

AIM - An integer variable which assumes the value 0 if any of the four conditions mentioned in the description above are not met. AIM assumes the value of BAIM if all conditions are met.

LOSE - An integer variable which assumes the value 1 if line of sight exists between A and his platoon leader. LOSE assumes the value 0 if line of sight does not exist.

R - A real variable which represents the range in meters from A to the platoon leader's FOE.

Coding and Brief Explanation

The coding for routine COMMO.PASS.TGT is shown below, followed by a brief explanation:

```
1   ROUTINE COMMO.PASS.TGT(A) YIELDING AIM

2   DEFINE A, AIM, BAIM, AND LOSE AS INTEGER VARIABLES

3   DEFINE R AS A REAL VARIABLE

4   LET BAIM=FOE(PLTLDR(A))

5   IF BAIM EQ 0

6   LET AIM=0

7   RETURN

8   ELSE

9   IF ALIVE.DEAD(BAIM)  EQ 1  LET AIM=0  RETURN  ELSE

10  CALL LOC(BAIM)

11  CALL LOS GIVEN X.CURRENT(A),Y.CURRENT(A),Z.CURRENT
(A),HTO(A),X.CURRENT(BAIM),

12  Y.CURRENT(BAIM),Z.CURRENT(BAIM),HTTGT(BAIM),P.V(*)

13  LET PCT.VIS=P.V(1)

14  IF PCT.VIS GT CRITICAL.VALUE

15  LET LOSE=1  JUMP AHEAD  ELSE  LET LOSE=0  HERE

16  IF LOSE EQ 0

17  LET AIM=0

18  RETURN
```

```
19  ELSE

20  CALL DIST(X.CURRENT(A),Y.CURRENT(A),X.CURRENT(BAIM),
Y.CURRENT(BAIM))

21                  YIELDING R

22  IF R GT OP.RNG(A)

23  LET AIM=0

24  RETURN

25  ELSE

26  LET AIM=BAIM

27  RETURN

28  END
```

Lines 1-3 define the routine and local variables.

Line 4 assigns to BAIM the pointer value of the platoon
leader's FOE.

Lines 5-8 test the platoon leader's FOE. If the
platoon leader has no FOE, AIM is set to 0 and control is
returned to routine T72.TACTICS.

Line 9 insures that BAIM is still alive before proceeding.
If not, AIM is set to 0 and control returns to routine
T72.TACTICS.

Line 10 updates the current X, Y, and Z coordinates.

Lines 11-15 check line of sight between the observer, A,
and his potential target, BAIM. If the percentage of the
target visible to the observer is greater than a user
defined threshold, line of sight exists.

Lines 16-18 return AIM with a value of 0 to routine
T72.TACTICS if line of sight does not exist.

Lines 19-21 calculate the range in meters between the
observer and the target.

Lines 22-24 return AIM with a value of 0 to routine
T72.TACTICS if the potential target is outside the observer's
maximum effective range.

Lines 25-27 return AIM with a value equal to BAIM to
routine T72.TACTICS if both the line of sight and range
tests are satisfied.

Y.   ROUTINE BMP.TACTICS

Description

Routine BMP.TACTICS is used by event TARGET.SELECT to
determine if a BMP's currently accessed threat is being
engaged by another company member. If so, an ANSWER = 1
is returned to event TARGET.SELECT and the threat in question

is no longer considered a potential target during the current TARGET.SELECT.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value of the BMP which is attempting to select a target.

B - An integer variable containing the pointer value of the currently accessed threat in the BMP's detected list.

ANSWER - An integer variable which is assigned a value of 1 if A's potential target, B, is being engaged by another company member.

Coding and Brief Explanation

The coding for routine BMP.TACTICS is shown below, followed by a brief explanation:

```
1  ROUTINE BMP.TACTICS(A,B)  YIELDING ANSWER
2  DEFINE A, B, AND ANSWER AS INTEGER VARIABLES
3  LET ANSWER = 0
4  FOR EACH TANK IN COMP.UNIT(CO(A  )), DO
5  IF B EQ FOE(TANK) LET ANSWER = 1 RETURN
6  ELSE LOOP RETURN END
```

Lines 1-2 define the routine and local variables.

Line 3 sets the default value of ANSWER to 0.

Lines 4-6 search other company members and determine if anyone is engaging A's potential target. The appropriate value for ANSWER is set and control returns to event TARGET.SELECT.

Z.   ROUTINE XM1.TACTICS

Description

Routine XM1.TACTICS is used by event TARGET.SELECT to determine if an XM1's currently accessed threat is being engaged by another platoon member. If this is the case, an ANSWER = 1 is returned to TARGET.SELECT and the threat in question is no longer considered a potential target during the current target selection event.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value of the TANK which is attempting to select a target.

B - An integer variable containing the pointer value of the currently accessed threat in the XM1's detected list.

ANSWER - An integer variable which is assigned a value of 1 if A's potential target, B, is being engaged by another platoon member.

Coding and Brief Explanation

The coding for routine XM1.TACTICS is shown below,
followed by brief explanation:

```
1  ROUTINE XM1.TACTICS(A,B) YIELDING ANSWER

2  DEFINE A, B, AND ANSWER AS INTEGER VARIABLES

3  LET ANSWER = 0

4  FOR EACH TANK IN PLT.UNIT(PLT(A  )), DO

5  IF B EQ FOE(TANK) LET ANSWER = 1 RETURN

6  ELSE LOOP RETURN END
```

Lines 1-2 define the routine and local variables.

Line 3 sets the default value of ANSWER to 0.

Lines 4-6 search other platoon members and determine
if anyone in the platoon is engaging A's potential target.
The appropriate value of ANSWER is set and control is
returned to event TARGET.SELECT.

AA.   ROUTINE IFV.TACTICS

Description

Routine IFV.TACTICS is used by event TARGET.SELECT to
determine if an IFV's currently accessed threat is being
engaged by another platoon member. If this is the case, an

ANSWER = 1 is returned to TARGET.SELECT and the threat in
question is no longer considered a potential target during
the current target selection event.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value of
the IFV which is attempting to select a target.

B - An integer variable containing the pointer value of
the currently accessed threat in the IFV's detected list.

ANSWER - An integer variable which is assigned a value
of 1 if A's potential target, B, is being engaged by another
platoon member.

Coding and Brief Explanation

The coding for routine IFV.TACTICS is shown below,
followed by a brief explanation:

```
1  ROUTINE ⁄IFV.TACTICS(A,B) YIELDING ANSWER

2  DEFINE A, B, AND ANSWER AS INTEGER VARIABLES

3  LET ANSWER = 0

4  FOR EACH TANK IN PLT.UNIT(PLT(A )), DO

5  IF B EQ FOE(TANK) LET ANSWER = 1 RETURN

6  ELSE LOOP RETURN END
```

Lines 1-2 define the routine and local variables.

Line 3 sets the default value of ANSWER to 0.

Lines 4-6 search other platoon members and determine
if anyone is engaging A's potential target. Control is
then returned to event TARGET.SELECT.

BB. ROUTINE ITV.TACTICS

<u>Description</u>

Routine ITV.TACTICS is identical to routine IFV.TACTICS.
The coding is shown below. A description, list of local
variables, and brief explanation of the coding is given in
routine IFV.TACTICS.

```
1  ROUTINE ITV.TACTICS(A,B) YIELDING ANSWER

2  DEFINE A, B, AND ANSWER AS INTEGER VARIABLES

3  LET ANSWER = 0

4  FOR EACH TANK IN PLT.UNIT(PLT(A  )), DO

5  IF B EQ FOE(TANK) LET ANSWER = 1 RETURN

6  ELSE LOOP RETURN END
```

CC. ROUTINE PRIORITY.AND.ROUND.SELECT

<u>Description</u>

Routine PRIORITY.AND.ROUND.SELECT is used by event
TARGET.SELECT to assign a priority to the threat currently

being accessed and to specify the type of ammunition to be used against this threat. As previously discussed, these priorities and ammunition selections are provided in the danger state arrays, DS1 and DS2 (see Tables III and IV in chapter V). Danger state arrays provided as part of the basic model are currently limited to five BLUE weapon types and three RED weapon types, with a maximum of four ammunition types each. Weapon system and ammunition codes are shown in table I.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value of the TANK for which a target priority and an ammunition selection is required.

B - An integer variable containing the pointer value of the currently accessed threat in the selecting TANK's.

ANSWER - The returned argument from routine AMMO.CHECK. An ANSWER = 1 indicates that the required ammunition is available. An ANSWER = 0 indicates the converse.

I - An integer counter which accesses the appropriate row of the danger state array.

J - An Integer counter which accesses the appropriate
column of the danger state array.

THRESHOLD - An integer constant used as an upper bound
for P.

P - An integer variable which assumes the priority
value in the currently accessed row and column of the danger
state array.

RND - The preferred ammunition type associated with the
target for which a priority has been assessed.

Coding and Brief Explanation

The coding for routine PRIORITY.AND.ROUND.SELECT is
shown below, followed by a brief explanation:

```
1   ROUTINE PRIORITY.AND.ROUND.SELECT(A,B) YIELDING P,
AND RND
2   DEFINE A,B,P,RND,I,J,ANSWER,AND THRESHOLD AS INTEGER
VARIABLES
3   LET THRESHOLD=99
4   IF COLOR(A) EQ BLUE
5   LET I=WPN.TYPE(A)
6   IF I EQ 1 LET I=2
7   ALWAYS LET I=I-1
8   GO TO I0, I3, I6, I9, I12   PER   I
```

```
9    'I0'   LET I=0   GO TO BANDS

10   'I3'   LET I=3   GO TO BANDS

11   'I6'   LET I=6   GO TO BANDS

12   'I9'   LET I=9   GO TO BANDS

13   'I12'   LET I=12   GO TO BANDS

14   'BANDS'   LET I=I+I+MIN.F(TRUNC.F(RANGE(A)/1000),2)

15   GO TO J1,J9,J17   PER WPN.TYPE(B)-6

16   'J1'   LET J=1   GO TO START

17   'J9'   LET J=9   GO TO START

18   'J17'   LET J=17   GO TO START

19   'START'

20   LET P=DS1(I,J)

21   IF P GE THRESHOLD RETURN ELSE

22   LET RND=DS1(I,J+1)

23   CALL AMMO.CHECK(A,RND) YIELDING ANSWER

24   IF ANSWER EQ 0   LET J=J+2   GO TO START

25   ELSE RETURN

26   ELSE

27   LET I=WPN.TYPE(A)-6

28   GO TO I10,I13,I16    PER I

29   'II0'   LET I=0   GO TO SBANDS

30   'II3'   LET I=3   GO TO SBANDS
```

```
31  'II6'  LET I=6  GO TO SBANDS

32  'SBANDS'  LET I=I+1+MIN.F(TRUNC.F(RANGE(A)/1000),2)

33  LET J=WPN.TYPE(B)

34  IF J=1  LET J=2

35  ALWAYS LET J=J-1

36  GO TO JJ1,JJ9,JJ17,JJ25,JJ33   PER J

37  'JJ1'  LET J=1  GO TO SSTART

38  'JJ9'  LET J=9  GO TO SSTART

39  'JJ17'  LET J=17  GO TO SSTART

40  'JJ25'  LET J=25  GO TO SSTART

41  'JJ33'  LET J=33  GO TO SSTART

42  'SSTART'

43  LET P=DS2(I,J)

44  IF P GE THRESHOLD  RETURN ELSE

45  LET RND=DS2(I,J+1)

46  CALL AMMO.CHECK(A,RND) YIELDING ANSWER

47  IF ANSWER EQ O  LET J=J+2  GO TO SSTART

48  ELSE RETURN

49  END
```

Lines 1-2 define the routine and local variables.

Line 3 sets the value of THRESHOLD to 99.

Line 4 accesses the BLUE portion of the routine.

157

Lines 5-7 calculate the firer's weapon type to insure
that it reflects one of the available weapon types in
the danger state array.

Lines 8-13 reset the row index, I, based on the weapon
type of the firer, A, and transfer control to label 'BANDS'.

Line 14 adds a minimum of 1 and a maximum of 3 to the
current value of I. This accesses the proper row of the
danger state array in accordance with the weapon type of
the firer and the range band of the target.

Lines 15-19 set an initial value for the column index,
J, based on the weapon type of the target.

Lines 20-21 calculate the target priority within the
given rangeband. If the priority equals or exceeds the
value of THRESHOLD, the target cannot be engaged and control
returns to event TARGET.SELECT.

Line 22 returns the type of ammunition required to engage
the target.

Lines 23-25 verify that the required ammunition is
available. If not the program transfers to line 19. If
the ammunition is available, the priority, P, and ammunition
type, RND, are returned to event TARGET.SELECT.

Line 26 accesses the RED portion of the routine.

Lines 28-31 set an initial value for the row index, I, based on the weapon type of the firer, and transfer control to label 'SBANDS'.

Line 32 adds a minimum of 1 and a maximum of 3 to the current value of I. This accesses the proper row of the danger state array in accordance with the weapon type of the firer and rangeband of the target.

Lines 33-41 set an initial value for the column index, J, based on the weapon type of the target.

Lines 42-44 calculate the target priority within the given rangeband. If the priority equals or exceeds the value of THRESHOLD, the target cannot be engaged and a return to event TARGET.SELECT is effected.

Line 45 returns the type of ammunition required to engage the target.

Lines 46-48 verify that the required ammunition is available. If not the program transfers to line 32. If the ammunition is available, the priority, P, and ammunition type, RND, are returned to event TARGET.SELECT.

DD. ROUTINE AMMO.CHECK

Description

Routine AMMO.CHECK is used by three other STAR routines:

159

T72.TACTICS , PRIORITY, AND ROUND.SELECT, AND WE.MISS. In all
cases a return of ANSWER = 0 from AMMO.CHECK indicates that
the ammunition of the type specified is not available. The
converse is true for ANSWER = 1.

If the required ammunition is available in T72.TACTICS,
the TANK is allowed to select as his FOE the FOE of his
platoon leader. The TANK then schedules a FIRE on this
FOE. If the required ammunition is not available, the TANK
initiates a normal target selection by returning to event
TARGET.SELECT.

Similarly, in routine PRIORITY.AND.ROUND.SELECT, a
lack of the required ammunition will cause a further search
of the danger state array.

Finally, a lack of the required ammunition in routine
WE.MISS initiates a target selection as opposed to firing
again on the same target. The lack of ammunition may cause
another target to assume a higher priority as determined
from the danger state array.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value of
TANK for which ammunition is being checked.

RND - An integer variable containing the numeric type

of ammunition being checked.

ANSWER - An integer variable which takes on the value 1

if ammunition of the type corresponding to RND is available,

and the value 0 if this type of ammunition is not available.

Coding and Brief Explanation

The coding for ROUTINE AMMO.CHECK is shown below,

followed by a brief explanation:

```
1  ROUTINE AMMO.CHECK(A,RND) YIELDING ANSWER

2  DEFINE A,RND,AND ANSWER AS INTEGER VARIABLES

3  LET ANSWER=1

4  GO TO AP,HE,AW1,AW2  PER RND

5  'AP'  IF AP.TOW(A) EQ 0 LET ANSWER=0  ALWAYS RETURN

6  'HE'  IF HE.DRAG(A) EQ 0 LET ANSWER=0  ALWAYS RETURN

7  'AW1'  IF AW1.OR.MSL3(A) EQ 0  LET ANSWER=0  ALWAYS
RETURN

8  'AW2'  IF AW2.OR.ADM(A) EQ 0  LET ANSWER=0  ALWAYS
RETURN

9  END
```

Lines 1-3 define the routine, local variables, and set

the variable ANSWER to a default value of 1.

Line 4 directs a transfer to one of four labels by means of a computed go to statement. RND may take on the values 1, 2, 3, or 4.

Lines 5-8 check the appropriate ammunition count.

EE. ROUTINE STOP.TO.FIRE

Description

Routine STOP.TO.FIRE is used by events FIRE and IMPACT to halt and restart missile firing weapon systems at appropriate times. In general, missile firing systems will stop propr to firing and begin movement after assessing results of their current round. If other than missile firing systems are required to use these tactics, the IF statement prior to CALL STOP.TO.FIRE in events FIRE and IMPACT must be changed to test for the added weapon system types.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value of the TANK for which a stop or start is required.

STOPCOUNT - An integer variable which assumes the value 1 if the routine is called from event FIRE, or the value 2 if the routine is called from event IMPACT.

162

<u>Coding and Brief Explanation</u>

The coding for routine STOP.TO.FIRE is shown below,

followed by a brief explanation:

```
1   ROUTINE STOP.TO.FIRE(A,STOPCOUNT)

2   DEFINE A AND STOPCOUNT AS INTEGER VARIABLES

3   IF PROJO(A) EQ 3 RETURN ELSE

4   GO TO X1,X2 PER STOPCOUNT

5   'X1'  LET SPD(A)=0.0

6   LET T.SPD(A)=TIME.V

7   RETURN

8   'X2'  LET SPD(A)=7.0

9   LET T.SPD(A)=TIME.V

10  LET SECOND.SHOT(A)=0

11  RETURN

12  END
```

Lines 1-2 define the routine and local variables.

Line 3 insures that control is returned to the calling

event if anything other than a missile (ammunition types 1 or

2) is being fired.

Line 4 transfers control to label X1 (a stop called from

event FIRE) if STOPCOUNT is 1 or to label X2 (a start called

from event IMPACT) if STOPCOUNT is 2.

163

Lines 5-7 set the firer's speed to zero, record the time that the speed was set to zero, and return control to event FIRE.

Lines 8-11 set the firer's speed to desired unit speed, record the time that this speed was set, reset the second shot counter attribute to zero, and return control to event IMPACT.

FF.   ROUTINE SET.MUZZLE.VEL

Description

Routine SET.MUZZLE.VEL is used by event FIRE to obtain the velocity of the type of ammunition being fired. The range to the target divided by this velocity yields the time of flight of the round. This time is used by event FIRE to schedule event IMPACT. As discussed in the PREAMBLE section, the M1, M2, M3, and M4 attributes for muzzle velocities are associated with projectile or ammunition types 1, 2, 3, or 4.

Local Variable

The following is a local variable of this routine:

A - An integer variable containing the pointer variable of the weapon system for which a muzzle velocity is required.

Coding and Brief Explanation

The coding for routine SET.MUZZLE.VEL is shown below.
As discussed previously, PROJO(A) is an integer variable
which assumes values 1 through 4 depending on the ammunition
type being fired.   This keys the computed go to statement to
the appropriate label.   The remander of the code is self
explanatory.

```
1  ROUTINE SET.MUZZLE.VEL(A)

2  DEFINE A AS AN INTEGER VARIABLE

3  GO TO AP, HE, AW1, AW2 PER PROJO(A)

4  'AP'  LET MZL.VEL(A) = M1(A) RETURN

5  'HE'  LET MZL.VEL(A) = M2(A) RETURN

6  'AW1'  LET MZL.VEL(A) = M3(A) RETURN

7  'AW2'  LET MZL.VEL(A) = M4(A) RETURN

8  END
```

GG.   ROUTINE DECREMENT.AMMO

Description

Routine DECREMENT.AMMO is used by event IMPACT to reflect
the expenditure of ammunition.   Any of four ammunition
categories defined by the user may be decremented.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value of the TANK for which ammunition is being decremented.

RND - An integer variable containing the numeric type of ammunition being decremented.

Coding and Brief Explanation

The coding for routine DECREMENT.AMMO is shown below, followed by a brief explanation:

```
1  ROUTINE DECREMENT.AMMO(A,RND)

2  DEFINE A AND RND AS INTEGER VARIABLES

3  GO TO AP,HE,AW1,AW2  PER RND

4  'AP'  LET AP.TOW(A)=AP.TOW(A)-1  RETURN

5  'HE'  LET HE.DRAG(A)=HE.DRAG(A)-1  RETURN

6  'AW1'  LET AW1.OR.MSL3(A)=AW1.OR.MSL3(A)-1  RETURN

7  'AW2'  LET AW2.OR.ADM(A)=AW2.OR.ADM(A)-1  RETURN

8  END
```

Lines 1-2 define the routine and local variables.

Line 3 directs a transfer to one of four labels by means of a computer go to statement. RND may take on the values 1, 2, 3, or 4.

Lines 4-7 decrement the appropriate ammunition type by
subtracting 1 from a TANK's ammunition type attribute.
Control is returned to event IMPACT.

HH.   ROUTINE COMPUTE

Description

Routine COMPUTE is used by event IMPACT to assign a
probability of hit for each weapon system and ammunition
type.   The values generated by routine COMPUTE are gross
approximations and are used solely to provide values which
will exercise the model.   Users should supply their own
hit probability routines.   Recall that in current version
of STAR, a hit is synonomous with a catastrophic kill.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value of
the TANK which has fired.

ID - An integer variable containing the pointer value of
the target TANK.

PCT.VIS - A real variable which may take on continous
values between 0 and 1.   Note that a PCT.VIS of less than
1 reduces the computed probability of hit.

Coding and Brief Explanation

The coding for routine COMPUTE is shown below, followed by a brief explanation:

1 ROUTINE COMPUTE(A,ID,PCT.VIS)

2 DEFINE A AND ID AS INTEGER VARIABLES

3 DEFINE PCT.VIS AND R AS REAL VARIABLES

4 CALL DIST(X.CURRENT(A),Y.CURRENT(A),X.CURRENT(ID), Y.CURRENT(ID)) YIELDING R

5 LET RANGE(A) = R

6 GO TO XM1,XM1,IFV,ITV,DIVAD,DRAGON,T72,BMP,ZSU PER WPN.TYPE(A)

7 'XM1' 'T72' LET PH(A)=1-RANGE(A)/3000. GO TO ZSU

8 'IFV' 'ITV' 'DRAGON' LET PH(A) = 1 - .15 * RANGE(A) / 3000.0 GO TO ZSU

9 'BMP' LET PH(A) = 1 - .25 * RANGE(A) / 3000.0 GO TO ZSU

10 'ZSU' 'DIVAD' IF PROJO(A)=3 LET PH(A)=.3 ALWAYS LET PH(A)=PCT.VIS*PH(A) RETURN

11 END

Lines 1-3 define the routine and local variables.

Line 4 calculates the current range to the target.

Line 5 sets the RANGE attribute of the firer to the current range calculated above.

Line 6 directs a transfer to one of 9 labels by means of a computed go to statement.

Lines 7-10 compute the appropriate probability of hit; control is then returned to event IMPACT.

II. ROUTINE TALLY.HIT.STATE

Description

Routine TALLY.HIT.STATE is used by event IMPACT to account for each type of kill sustained by a weapon system. The following categories are available: ND.HIT, M.KILL, F.KILL, MF.KILL, and K.KILL. These categories are weapon system attributes and are defined in the discussion of the PREAMBLE. In addition, this routine accumulates the total number of hits sustained and the total number of rounds fired at the weapon system (hits plus misses). Currently, only the 'K.KILL' and 'MISS' sections of this routine are used.

Local Variables

The following are local variables of this routine:

TANK - An integer variable containing the pointer value of the weapon system for which a hit category is being updated.

169

DAMAGE.NUM - An integer variable which takes on the
values 1 through 6 depending on the type of hit sustained.

Coding and Brief Explanation

The coding for routine TALLY.HIT.STATE is shown below,
followed by a brief explanation:

```
 1  ROUTINE TALLY.HIT.STATE(TANK,DAMAGE.NUM)

 2  DEFINE TANK AND DAMAGE.NUM AS INTEGER VARIABLES

 3  GO TO NO.DAMAGE,M.KILL.F.KILL,MF.KILL,K.KILL,MISS
PER DAMAGE.NUM

 4  'NO.DAMAGE'

 5  LET ND.HIT(TANK)=MD.HIT(TANK)+1

 6  LET NUM.HIT(TANK)=NUM.HIT(TANK)+1

 7  LET FIRED.AT(TANK)=FIRED.AT(TANK)+1

 8  'M.KILL'

 9  LET M.HIT(TANK)=M.HIT(TANK)+1

10  LET NUM.HIT(TANK)=NUM.HIT(TANK)+1

11  LET FIRED.AT(TANK)=FIRED.AT(TANK)+1

12  RETURN

13  'F.KILL'

14  LET F.HIT(TANK)=F.HIT(TANK)+1

15  LET NUM.HIT(TANK)=NUM.HIT(TANK)+1

16  LET FIRED.AT(TANK)=FIRED.AT(TANK)+1
```

```
17   RETURN

18   'MF.KILL'

19   LET MF.HIT(TANK)=MF.HIT(TANK)+1

20   LET NUM.HIT(TANK)=NUM.HIT(TANK)+1

21   LET FIRED.AT(TANK)=FIRED.AT(TANK)+1

22   RETURN

23   'K.KILL'

24   LET K.HIT(TANK)=K.HIT(TANK)+1

25   LET NUM.HIT(TANK)=NUM.HIT(TANK)+1

26   LET FIRED.AT(TANK)=FIRED.AT(TANK)+1

27   IF COLOR(TANK) EQ BLUE AND M.BLUE.ALIVE(TANK) EQ 1
REMOVE TANK FROM BLUE.ALIVE

28   REMOVE TANK FROM PLT.UNIT(PLT(TANK))

29   REMOVE TANK FROM COMP.UNIT(CO(TANK))

30       RETURN ELSE

31   IF M.RED.ALIVE(TANK) EQ 1 REMOVE TANK FROM RED.ALIVE

32   REMOVE TANK FROM PLT.UNIT(PLT(TANK)) REMOVE TANK FROM
COMP.UNIT(CO(TANK))

33   ALWAYS RETURN

34   'MISS'

35   LET FIRED.AT(TANK)=FIRED.AT(TANK)+1

36   RETURN

37   END
```

Lines 1-2 define the routine and local variables.

Line 3 uses a computed go to statement to access the appropriate hit category.

Lines 4-7 increment by one the number of no damage hits sustained, the total number of hits sustained, and the total number of rounds fired at the weapon system. Control is then returned to event IMPACT.

Lines 8-12 are similar to those above except that the number of mobility hits sustained is incremented by one.

Lines 13-17 are similar to those above except that the number of firepower hits sustained is incremented by one.

Lines 18-22 are similar to those above except that the number of mobility and firepower hits are incremented by one.

Lines 23-26 are similar to those above except that the number of catastrophic kills is incremented by one. Control is not returned to event IMPACT until set manipulation is accomplished by the next 7 lines.

Lines 27-30 insure that the TANK is BLUE and that he has not already been removed from the set of blue, alive tanks. If these conditions are met the TANK is removed from sets

Coding and Brief Explanation

The coding for routine WE.MISS is shown below, followed
by a brief explanation:

1   ROUTINE WE.MISS(A)

2   DEFINE A AND ANSWER AS INTEGER VARIABLES   DEFINE R
AS A REAL VARIABLE

3   LET MISSHOT(A) = MISSSHOT(A) + 1

4   GO TO XM1,XM1,IFV,ITV,DIVAD,DRAGON,T72,BMP,ZSU PER
WPN.TYPE(A)

5   'XM1'   'DIVAD'

6   IF MISSSHOT(A) EQ 2 OR HITSHOT(A) EQ 2

7   'MOVE'

8   LET HITSHOT(A) = 0 LET MISSSHOT(A) = 0 LET FOE(A) = 0

9   SCHEDULE A HIDE(A,1) NOW

10   RETURN ELSE

11   CALL AMMO.CHECK(A,PROJO(A)) YIELDING ANSWER

12   IF ANSWER EQ 0 GO TO ZSU

13   ELSE GO TO FIRE

14   'IFV'

15   'ITV'

16   'DRAGON'

17   IF HITSHOT(A) + MISSSHOT(A) GE 2

173

18 GO TO MOVE ELSE GO TO T72

19 'T72'

20 'BMP'

21 'ZSU'

22 LET FOE(A) = 0 SCHEDULE A TARGET.SELECT(A) NOW

23 RETURN

24 'FIRE'

25 CALL DIST(X.CURRENT(A),Y.CURRENT(A),X.CURRENT(FOE)),
Y.CURRENT(FOE(A)))

26 YIELDING R

27 LET RANGE(A) = R LET SCHED(A) = 1

28 SCHEDULE A FIRE(A,FOE(A)) IN UNIFORM.F(4.,10.,8) UNITS

29 LET CHECK.TIME(A) = TIME.A(FIRE)

30 RETURN END

Lines 1-2 define the routine and local variables.

Line 3 increments by one the MISSSHOT attribute of
the firer.

Line 4 directs a transfer to the appropriate label by
means of a computed go to statement. The weapon type of
the firer keys this transfer.

Line 5 labels the XM1 and DIVAD section of the routine.

Lines 6-9 test the status of the HITSHOT and MISSSHOT attributes of the firer. Two misses or two hits (K-kills) cause the hit and miss counters and FOE attributes of the firer to be reset to 0. A movement to a full defilade position is accomplished and control is returned to event IMPACT.

Lines 10-11 are executed if the hit and miss test above fails. These lines insure that the required ammunition is available prior to firing again on the same target.

Lines 12-14 label the IFV, ITV, and DRAGON section of the routine.

Lines 15-16 count the number of rounds fired from the same position. If this sum is greater than or equal to 2, then hit and miss counters are reset to 0, the FOE attribute of the firer is reset to 0, a movement to defilade is accomplished and control is returned to event IMPACT.

Lines 17-19 label the T72, BMP, and ZSU section of the routine.

Lines 20-21 reset the FOE attribute of the firer to 0, and schedule an immediate TARGET.SELECT. This applies to all weapons of type T72, BMP, and ZSU. It also applies to

Lines 31-33 are similar to lines 27-30 except that they apply to RED TANKs.

Lines 34-36 increment by one the total number of rounds fired at the TANK and return control to event IMPACT.

JJ.   ROUTINE WE.MISS

Description

Routine WE.MISS is used by event IMPACT when anything other than a catastrophic kill (K-kill) is registered by the firer. Depending on the number of rounds fired and the weapon type of the firer, routine WE.MISS causes a movement to defilade, a target selection, or another engagement of the current target.

Local Variables

The following are local variables of this routine:

A - An integer variable containing the pointer value of the TANK for which a movement to defilade, target selection, or second engagement is required.

ANSWER - The returned argument from routine AMMO.CHECK. An ANSWER=1 indicates that the required ammunition is available. An ANSWER = 0 indicates the converse.

weapons of type XM1 and DIVAD which do not have the

ammunition type required to re-engage their current FOE.

Control is returned to event IMPACT.

Lines 22-27 calculate the current range to the target,

schedule a FIRE event in the appropriate amount of time,

and set the CHECK.TIME attribute of the firer to the time

for which the fire is scheduled. Control is then returned

to event IMPACT.

KK. ROUTINE WE.HIT

Description

Routine WE.HIT is used by event IMPACT when a

catastrophic kill (K-kill) is inflicted by the firer. Depend-

ing on the number of rounds fired previously and the weapon

type of the firer, routine WE.HIT causes a movement to

defilade (event HIDE) or another target selection. A

discussion of specific firing tactics for each weapon

system was presented in chapter III.

Local Variables

The following is a local variable of this routine:

A - An integer variable containing the pointer value of

the TANK for which a movement to defilade or target

selection is required.

177

Coding and Brief Explanation

The coding for routine WE.HIT is shown below, followed by a brief explanation:

```
 1  ROUTINE WE.HIT(A)

 2  DEFINE A AS AN INTEGER VARIABLE

 3  LET HITSHOT(A) = HITSHOT(A) + 1

 4  GO TO XM1,XM1,IFV,ITV,DIVAD,DRAGON,T72,BMP,ZSU   PER
WPN.TYPE(A)

 5  'XM1'    'DIVAD'

 6  IF (HITSHOT(A) EQ 2) OR (MISSSHOT(A) EQ 2) OR

 7  (HITSHOT(A) EQ 1 AND MISSSHOT(A) EQ 1)

 8  'MOVE'

 9  LET FOE(A) = 0

10  LET HITSHOT(A) = 0 LET MISSSHOT(A) = 0

11  SCHEDULE A HIDE(A,1) NOW

12  RETURN ELSE GO TO ZSU

13  'IFV'

14  'ITV'

15  'DRAGON'

16  IF HITSHOT(A) + MISSSHOT(A) GE 2

17  GO TO MOVE ELSE GO TO T72

18  'T72'
```

19   'BMP'

20   'ZSU'

21   SCHEDULE A TARGET.SELECT(A) NOW RETURN

22   END

Lines 1-2 define the routine and local variable.

Line 3 increments the HITSHOT attribute of the firer
by one.

Line 4 directs a transfer to the appropriate label by
means of a computed go to statement. The weapon type of the
firer keys this transfer.

Line 5 labels the XM1 and DIVAD section of the routine.

Lines 6-10 test the status of the HITSHOT and MISSSHOT
attributes of the firer. Two misses, two hits (K-kills),
or a miss followed by a hit cause the hit and miss counters
to be reset to 0, effect a movement to defilade and return
control to event IMPACT.

Lines 11-13 label the IFV, ITV and DRAGON sections of
the routine. They also serve as a transfer point if lines
6 and 7 are not true.

Line 14 counts the sum of the number of rounds fired
from the current position. If this sum is greater than
or equal to 2, the hit and miss counters are reset to 0,

179

program flow is transferred to line 9, a movement to

defilade is effected and control is returned to event

IMPACT.

Lines 16-18 label the T72, BMP, and ZSU section of the

routine. These lines also serve as a transfer point if line

14 is not true.

Line 19 schedules an immediate TARGET.SELECT for weapons

of type T72, BMP, and ZSU. Moreover, if BLUE weapon systems

have failed their hit and miss counter tests, a TARGET.SELECT

is also scheduled for these weapon systems.

V. INPUT REQUIREMENTS

A. SIMSCRIPT INPUT

General

All SIMSCRIPT input to STAR is in free field card input; that is, each data value must be separated by a blank space. This is the only formatting requirement for SIMSCRIPT input. A sample card or cards follows the discussion of each set of input values described below.

R.NUM.ALIVE, B.NUM.ALIVE, and DELTA.T

R.NUM.ALIVE and B.NUM.ALIVE are integer values equal to the number of RED and BLUE TANKs to be created. DELTA.T is a real variable whose value is the time in seconds between detection scheduling events (event STEP.TIME). These values could appear on a card as:

162 77 15.0

which represent 162 RED TANKs, 77 BLUE TANKs, and a DELTA.T of 15 seconds.

PNUM and CNUM

PNUM and CNUM refer to the number of platoon and company sets to be created. These values could appear on a data card as:

63 15

which represent 63 platoons and 15 companies.

X.STOP, Y.STOP, R.PCT.ATT, and B.PCT.ATT

X.STOP and Y.STOP are real variables which are the coordinates at which RED TANKs are halted. The battlefield is a 10 kilometer by 10 kilometer square and the values for

these coordinates may range from 0.0 to 10000.0. R.PCT.ATT
and B.PCT.ATT are the attrition percentages for RED and BLUE
forces, which, if equaled or exceeded, will stop the
simulation. Both are real variables ranging from 0.0 to
1.0. These 4 values might appear on a date card as:

2500.0 2000.0 .90 .80

which represent an X.STOP coordinate of 2500.0, a Y.STOP
coordinate of 2000.0, a RED attrition point of 90 percent,
and a BLUE attrition point of 80 percent.

DS1 and DS2

Each force has a danger state array: DS1 for BLUE
forces and DS2 for RED forces. The rows of each array are
keyed to the weapon type of the firer and the range band of
the target (less than 1000 meters, between 1000 and 2000
meters, and greater than 2000 meters). Consequently, each
firer will have three rows, one for each range band. The
columns are keyed to the number of target types available.
Each target type has a set of 8 columns assigned to it. A
macro look at DS1 would reveal a format similar to the one
below where each block (*) represents 3 rows and 8 columns.

TARGET

	T72	BMP	ZSU
XM1	*		
IFV			
ITV			
DIVAD			
DRAGON			

The total dimension of this array is (5 x 3) by (3 x 8) or
360 values.

A micro look at the XM1-T72 block (*) would show:

```
                        T72
          1        1     2     2     99    99    99    99
XM1      21        1    22     2     99    99    99    99
         41        1    43     2     99    99    99    99
```

Consider the first row which represents the less than
1000 meter rangeband. The numbers in the columns are to be
read in pairs. For example, the first and second numbers in
row 1 form the pair (1,1) and mean that the XM1's first
priority in this rangeband is a T72, and the preferred
ammunition for use against the T72 is type 1, APDS. The
T72 is also second priority within this rangeband using
ammunition type 2, HEAT.

Looking at the second row, the 1000-2000 meter rangeband,
reveals a similar ordering and ammunition selection. The
T72 is still first and second priority within the rangeband,
a constant value of 20 being added to priorities in the
second row. Any constant could be added to the basic
priority value in a row, but the resulting priority number
must be larger than the largest priority number in a prior
row associated with that firer. An exception to this rule
are the entries with value 99. These values are used as
stopping values when routine PRIORITY.AND.ROUND.SELECT
accesses a danger state array. When a priority of 99 is

retrieved from the array, this indicates that the target type should not be engaged.

In the third rangeband, the T72 is again first priority with ammunition type 1, APDS, and then drops to third priority with ammunition type 2, HEAT. An examination of the complete set of danger state array rows and columns associated with the XM1 (Table III) provides an explanation of this priority situation.

In row 3 (the greater than 2000 meter rangeband) the BMP assumes second priority using ammunition type 2, HEAT.

Note in row 1 that ammunition type 3 (.50 caliber machingun for the XM1) is used against the BMP and ZSU. Use of ammunition type 3 is restricted to targets in this range band only. This reflects the relatively short effective range of small caliber ammunition.

The remaining rows and columns of DS1 are constructed in a similar manner. DS2 is formed the same way except the RED force firers are used in constructing rows and BLUE force targets are used in constructing columns. DS1 and DS2 are shown in their entirety in tables IV and V.

To input these values, the user may type each row on one or more cards. Each value is an integer separated from the next value one or more spaces.

The first 2 rows of DS1 as prepared for input in the current version of STAR are shown below: In this case each 2 cards represent 1 row of the DS1 array.

TABLE III

XM1 PORTION - DS1

	1	1	2	99	99	99	3	2	4	1	5	3	99	99	6	1	7	2	8	3	99	99	99
XM1	1	1	2	99	99	99	3	2	4	1	5	3	99	99	6	1	7	2	8	3	99	99	99
	21	1	22	2	99	99	99	23	2	24	1	99	99	99	25	1	26	2	99	99	99	99	99
	41	1	43	2	99	99	99	42	2	44	1	99	99	99	45	1	46	2	99	99	99	99	99

185

TABLE IV

BLUE DANGER STATE ARRAY - DS1

```
 1  1  2  2 99 99 99  3  2  4  1  5  3 99 99  6  1  7  2  8  3 99 99
21  1 22  2 99 99 99 23  2 24  1 99 99 99 99 25  1 26  2 99 99
31  1 33  2 99 99 99 32  2 34  1 99 99 99 99 35  1 36  2 99 99
 3  1  1 99 99 99 99  1  3  2  1 99 99 99 99  4  3  5  1 99 99 99
21  1  1 99 99 99 99 22  1 23  3 99 99 99 24  1 25  3 99 99 99
31  1  1 99 99 99 99 32  1 99 99 99 99 99 33  1 99 99 99 99
 1  1  1 99 99 99 99  2  1 99 99 99 99 99  3  1 99 99 99 99 99
21  1  1 99 99 99 99 22  1 99 99 99 99 99 23  1 99 99 99 99
31  1  1 99 99 99 99 32  1 99 99 99 99 99 33  1 99 99 99 99
 8  8 99 99 99 99 99  2  3 99 99 99 99 99  1  3 99 99 99 99 99
99 99 99 99 99 99 99 99 99 99 99 99 99 99 21  3 99 99 99 99
 1  2 99 99 99 99 99  2  2 99 99 99 99 99  3  2 99 99 99 99 99
99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99
99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99
```

TABLE V

RED DANGER STATE ARRAY - DS2

```
1  1  2  2 99 99 99  3  2  5  1  9  3 99 99 99  4  2  6  1 10  3 99 99 11  1 12  2 99 99 99
7  2  8  3 99 99 99
21  1 22  2 99 99 99 24  2 26  1 31  3 99 99 23  2 25  1 30  3 99 99 27  2 28  1
99 99 99  2 99 99 99 29  3 99 99 99 99 99
41  1 44  2 99 99 99 43  2 46  1 99 99 99 42  2 45  1 99 99 99 47  2 48  1 99
99 99 99 99 99 99 99 99 99
2  1 99 99 99  1  3  3  1 99 99 99  4  3  5  1 99 99 99  6  1  7  3 99 99 99
99  8  3  9  1
21  1 99 99 99 23  1 25  3 99 99 22  1 24  3 99 99 26  1 27  3 99
99 99 28  3 99 99 99
41  1 99 99 99 43  1 99 99 99 42  1 99 99 99 44  1 99 99
99 99 99 99 99 99 99  2  3 99 99 99  3  3 99 99 99  1  3 99 99 99
99 99 99 99 99 99 99 99 99 99
99 99 99  4  3 99 99 99 23  3 99 99 99 22  8 99 99 99 21  3
99 99 99 99 99 99 99 99 99 99 99 99 99 99 99
99 99 99 99 99 99 99 99 99 99 99 99 99 99 99
99 99 99 99 99 99 99 99 99 99 99 99 99 99 99
99 99 99 99 99 99 99 99 99 99
99 99 99 99 99 99 99 99 99 99
```

```
CARD 1     1  1  2  2 99 99 99 99  3  2  4  1
CARD 2     5  3 99 99  6  1  7  2  8  3 99 99
CARD 3    21  1 22  2 99 99 99 99 23  2 24  1
CARD 4    99 99 99 99 25  1 26  2 99 99 99 99
```

Attributes

A number of attributes of the temporary entities, TANKs, must be input on data cards. In the discussion that follows, real variables will be denoted by an R in parentheses follow-the attribute name. Real attributes may be input with or without decimal points. Integer attributes will be denoted by an I-xxxx in parentheses following the attribute name. The xxxx sequence represents the maximum numerical value that the attribute may be assigned. These maximum sizes must be strictly observed since many of the integer attributes are bit packed (see the discussion of the PREAMBLE). Integer attributes may not be input with decimal points.

NAME(I-1023) - The first TANK has the name 1 with subsequent TANKs numbered sequentially and the last BLUE TANK assigned a NAME equal to the value of B.NUM.ALIVE. RED TANKs begin with a NAME equal to B.NUM.ALIVE + 1, and end with a NAME equal to the value of R.NUM.ALIVE.

COLOR(I-1) - A value of 1 indicates a BLUE TANK. A value of 0 denotes a RED TANK.

WPN.TYPE(I-31) - The nine weapon types and their codes are shown below:

1 - XM1(105mm main gun)

        2 - XM1(120mm main gun)

        3 - IFV

        4 - ITV

        5 - DIVAD

        6 - DRAGON

        7 - T72

        8 - BMP

        9 - ZSU

VEH.TYPE(I-31) - The 12 weapon types and their codes
are shown below:

        1 - XM1(105mm main gun)

        2 - XM1(120mm main gun, 40 round configuration)

        3 - XM1(120mm main gun, 43 round configuration)

        4 - XM1(120mm main gun, 49 round configuration)

        5 - XM1(120mm main gun, 52 round configuration)

        6 - IFV

        7 - ITV

        8 - DIVAD

        9 - DRAGON

       10 - T72

       11 - BMP

       12 - ZSU

AP.TOW(I-63) - Ammunition type 1. This represents the
initial amount on board. Table I provides a guide to the
specific type of ammunition that this and the next three

ammunition types represent with respect to a specific weapon system. If an ammunition type is not carried, a 0 should be entered.

HE.DRAG(I-63) - Ammunition type 2.

AW1.OR.MSL3(I-63) - Ammunition type 3.

AW2.OR.ADM(I-63) - Ammunition type 4.

SEC(I-2047) - The section number assignment. Since this organizational attribute has no effect on execution of STAR, any desired numbering scheme may be used.

PLT(I-2047) - The platoon number assignment. It is important for fire control purposes that TANKs be grouped sequentially in platoons and that TANKs in headquarters elements are assigned to the same platoon. The TANKs in a battalion headquarters, for example, are considered a platoon for fire control purposes and should all have the same platoon number assigned to them.

CO(I-1023) - The company number assignment. The comment above with respect to headquarters elements also applies in making company number assignments.

BN(I-127) - The battalion number assignment. Any numbering scheme may be used.

RGT(I-255) - The regiment number assignment. Any numbering scheme may be used. BLUE forces should have a 0 input.

BDE(I-7) - The brigade number assignment. Any numbering scheme may be used. RED forces should have a 0 input.

DIV(I-7) - The division number assignment. Any numbering scheme may be used.

X.CURRENT(R) - The TANK's initial X coordinate on the battlefield.

Y.CURRENT(R) - The TANK's initial Y coordinate on the battlefield.

SPD(R) - The TANK's initial speed in meters per second.

DIR.OF.MVMT(R) - The TANK's initial direction of movement in mils. An angle measured clockwise from East is negative. An angle measured counterclockwise from East is positive. Angles are entered in mils and converted to radians for use in the program.

PRI.DIR(R) - The primary direction of search measured in the same manner as DIR.OF.MVMT.

M1, M2, M3, M4(all I-65535) - The muzzle velocities in meters per second associated with each of the four ammunition types carried by the TANK. If an ammunition is not carried by a TANK, the associated velocity should be input as 0.

COCDR(I-9999999) - The NAME of the TANK's company commander. If the TANK is to be the company commander, the TANK's own NAME SHOULD be entered here and also for the PLTLDR and SECLDR attributes described below.

PLTLDR(I-9999999) - The NAME of a TANK's platoon leader. If the TANK is to be the platoon leader, the TANK's own NAME should be entered here and also for the SECLDR attribute described below.

191

SECLDR(I-9999999) - The NAME of a TANK's section leader.
If the TANK is to be the section leader, the TANK's own
name should be entered here.

OP.RNG(I-4095) - The TANK's maximum opening range in
meters.

A complete example of attribute input is shown in table
VI. A brief example showing the relationship of the NAME,
organization and leader attributes is shown below:

NAME	SEC	PLT	CO	COCDR	PLTLDR	SECLDR
1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	2	2	1	1	3	3
4	2	2	1	1	3	3
5	2	2	1	1	3	3
6	3	2	1	1	3	6
7	3	2	1	1	3	6
8	4	3	1	1	8	8
9	4	3	1	1	8	8
10	4	3	1	1	8	8
11	5	3	1	1	8	11
12	5	3	1	1	8	11
13	6	3	2	13	13	13
14	6	4	2	13	13	13
15	7	5	2	13	13	15
16	7	5	2	13	15	15
17	7	5	2	13	15	15
18	8	5	2	13	15	18

19	8	5	2	13	15	18
20	9	6	2	13	20	20
21	9	6	2	13	20	20
22	9	6	2	13	20	20
23	10	6	2	13	20	23
24	10	6	2	13	20	23

B. FORTRAN INPUT

FORTRAN input will not be discussed in this thesis since it is assumed that most readers are familiar with entering FORTRAN data. FORTRAN input required by the terrain model used in the current version of STAR may be obtained from Professor James K. Hartman, Naval Postgraduate School.

TABLE VI

SAMPLE ATTRIBUTE INPUT VALUES

INPUT SHOWN BELOW IS IN THE FOLLOWING ATTRIBUTE ORDER WITH 4 DATA CARDS PER TANK

```
AW2.OR.NAME    COLOR    WPN.TYPE    VEH.TYPE    AP.TOW    HE.DRAG    AW1.OR.MS
        CIV      SEC       PLT         CC          BN       RGT        BDE
    X.CURRENT  Y.CURRENT             SPD       DIR.OF.MV   PRI.DIR      MI
       M2        M3        M4        COCDR      PLTLDR     SECLDR     OP.RNG
```

NOTE THAT THE AW1.OR.MSL3 AND DIR.OF.MVMT ATTRIBUTE NAMES HAVE BEEN TRUNCATED
FOR COLUMN SPACING AND HEADING PURPOSES. THE FOLLOWING SAMPLE DATA IS FOR A
SIMULATION OF 31 BLUE TANKS VERSUS 53 RED TANKS:

```
  1      1       2        2 1 0      24 1     16        100 1
  0                                                     1
  1      1       5500     2          24       0         1400
  0      4540             1 0        1        0         2600
1200     2000   0         0          500      1         1
         1       2        2          1        16        100
  0      1                1 0                 0         1
  2      4970    5360     0          24       1         1400
  0              4        1 0        210      0         2600
1200     2000   0         0          1        1         1
  0      1       2        4          0        0         0
  3                       1 0                 0         0
  0      4910    5480     0          500      1         1
  1      0       2        1 0        24       16        100
  0      1                0                   0         1
  4      1       2        2 2        711      4         1400
  0      5300    5690     0          1        0         2600
1200     2000   0         1 0        24       16        1
  5      1                                    0         100
  0      1       2        2 2        408      4         1400
  1              5420     0          24       0         2600
  0      5400    0        1 0        622      16        1
  6      2000   2         1          4                  0
  0      1       2        2          24       0         1400
1200     5365            1 0        1 0       7         2600
  7      2000   5500      1          942                1
  0      1 2    0 2                  4                  1400
  1      5300   2         1                             2600
  0      2000   5820
1200            0
```

TABLE VI (CONTINUED)

```
  100        16      24      2      2          1            8
    1      0007    1333    1 0     5860       2           1
 1400        17      24      1 0                5200        0
 2600         1      24      2 0     3        2000      1200
  100        16    1085    1 0       5260      1           0
    1      0009      24    2 0        3         0          9
 1400        16      24    1 1                 5580        1
 2600         1     76C     2 0      2        2000         0
  100        16      24    1 0       5270      1        1200
    1      0009    5009    2 0         3        1           0
 1400        16      24    1 0                 5710        1
 2600         1      24    2 0       3         2000     1200
  100        16    1659    1 1                 2         11
    1      0009      24    1 0       5120        0        110
 1400        16      24    1 0        3        5760        1
 2600         1     -16    1 0       2         2000     1200
  100        16       9    1 1       5030      1         12
    1      0002      24    1 0        3          2        120
 1400        11    700     1 0                 5800        1
 2600         1      14    1 0      3          2000     1200
  100        12       1    1 0      300         1         13
    1      0004    145     1 0        3        5850        1
 1400         1      14    1 0       4         2000     1200
 2600         1       6    1 0      300         2         14
  100        16      14    1 0        4         0        140
    1      0004    504     1 0                 5090        1
 3200         1      14    1 0       4         2000       15
  100         1       6    1 0     4680         1          1
    1      0006    140     1 0        3        5150       16
 3200         1    -500    1 0        4        2000        1
  100        16      14    1 0     4540         2         17
    1      2000      14    1 0        6          0          1
 3200        14      10    1 0        4        5120       18
  100                  1     1 0    4830       2000        1
 3200                 14    1 0                5100      200
    0                              5840          1
    1
 1000
```

195

TABLE VI (CONTINUED)

15 0 1	1 5790	6 4740	6 1 0	0 1 -225	2 0 0	0 1 1000
200 0 1	2 5670	4 4680	16 1 0	14 16 20	14	0 1 1000
200 0 1	1 5000	6 5100	17 1 0	14 18 220	200 16	0 1 3000
210 0 1	1 4520	4 5100	17 1 0	218 745	0 0 0 21	0 1 3000
220 0 1	2 4700	5 5450	17 1 0	218 840	0 0 0 21	0 1 3000
230 0 1	2 4650	4 5430	17 1 0	238 1365	230 0 0	0 1 3000
240 0 1	1 4370 2000	3 5540	16 1 0	236 1735	230 0 0	1 2000 3100
250 0 1	2 4530 2000	3 5570	16 1 0	256 1425	250 0 0	0 1 3200 3100
260 0 1	2 4640 2000	3 5630	16 1 0	256 1180	250 0 0	0 1 3000 3100
270 0 1	1 4760 2000	6 5600	16 1 0	256 935	270 0 0	0 1 3200 3100
290 0 1	1 4430	0 5640	1 0 1	250 1720	270 0 0	0 1 3200 3000
200	0	0		25	25	1 0 1000

TABLE VI (CONTINUED)

TABLE VI (CONTINUED)

41	0	0	7 10	9 3 5	18	1 0 0
1300	6780	8120	3 4	-2400 3 5	22 1	1600
42	2000	0	7 10	3 4	35 18	2600 1 0 0
1300	6830	8070	3 5	-2400 3 5	39 22 1	1600
43	2000	0	7 11	3 4	35 18 1	2600 1 0 0
1300	6500	8000	3 4	-2400 4 3	43 22 1	1600
44	2000	0	7 11	3 5	43 18 1	2600 1 0 0
1300	6520	7950	3 4	-2400 4 3	43 22 1	1600
45	2000	0	7 11	3 5	43 18 1	2600 1 0 0
1300	6970	7920	3 4	-2400 4 3	43 22 1	1600
46	2000	0	7 11	3 5	43 18 1	2600 1 0 0
1300	7000	7860	3 4	-2400 4 7	43 1	1600
47	2000	0	8 12	3 5	47 18 1	2600 4 0 0
1300	6580	8580	3 4	-2400 4 7	47 1	3000
48	1600	0	8 12	3 5	47 18 1	2000 4 0 0
0	6660	8500	3 4	-2400 4 7	47 1	3000
49	1600	0	8 12	3 5	47 18 1	2000 4 0 0
1000	6800	8350	3 4	-2400 4 7	47 1	3000
50	1600	0	8 12	3 4	47 18 1	2000 4 0 0
1	6880	8250	3 5	-2400 4 7	22 1	3000
51	1600	0	7 13	4 5	51 11 1	2000 1 0 0
1300	7450 2000	7850 0	51	51	-2400 51	1600 2600

TABLE VI (CCNTINUED)

100 0 1600 2600	22 1 0 52	18 1 -2400 52	5 4 51 9	7 14 7820 0	6090 2000 0	52 0 1 1300
100 0 1600 2600	22 1 0 52	18 1 -2400 52	5 4 51 9	7 14 7775 0	7075 2000 0	53 0 1 1300
2100 0 1600 2600	22 1 0 52	18 1 -2400 52	5 4 51 9	7 14 7700 0	7140 2000 0	54 0 1 1300
100 0 1600 2600	22 1 0 52	18 1 -2400 52	5 4 51 9	7 14 7675 0	7170 2000 0	55 0 1 1300
2100 0 1600 2600	22 1 0 56	18 1 -2400 56	5 4 51 9	7 15 7630 0	7220 2000 0	56 0 1 1300
100 0 1600 2600	22 1 0 56	18 1 -2400 56	5 4 51 9	7 15 7550 0	7290 2000 0	57 0 1 1300
2100 0 1600 2600	22 1 0 56	18 1 -2400 56	5 4 51 9	7 15 7480 0	7325 2000 0	58 0 1 1300
100 0 1600 2600	22 1 0 56	18 1 -2400 56	5 4 51 9	7 15 7400 0	7420 2000 0	59 0 1 1300
2100 0 1600 2600	22 1 0 60	18 1 -2400 60	5 4 51 9	7 16 7340 0	7500 2000 0	60 0 1 1300
100 0 1600 2600	22 1 0 60	18 1 -2400 60	5 4 51 9	7 16 7305 0	7540 2000 0	61 0 1 1300
1600 2600	22 1 0 6	18 1 -2400 60	5 4 51	7 16 7275 0	7580 2000 0	62 0 1 1300

TABLE VI (CONTINUED)

100	22	18	9	7	0	63
0	1	-2400	4	16	0	0
1600	0	60	51	7200	7660	1
2600	0	8	10	0	2000	1300
40	1	1	4	8	0	64
0	60	-2400	51	17	0	0
2000	0	64	51	7760	7360	1
3000	1	3	10	0	1600	1300
40	0	1	4	8	0	65
0	64	-2400	51	17	0	0
2000	0	64	51	7700	7430	1
3000	1	8	10	0	1600	
40	0	1	4	8	0	66
0	64	-2400	51	17	0	0
2000	0	64	51	7570	7540	1
3000	1	8	10	0	1600	
40	0	1	4	8	0	67
0	64	-2400	51	17	0	0
2000	22	64	51	7500	7600	1
3000	0	8	9	0	1600	
10	1	1	5	7	1	68
0	68	-2400	68	18	0	0
1600	22	68	6	6870	8220	1
2600	0	1	9	0	2000	1300
10	1	1	5	7	0	69
0	65	-2400	51	19	0	0
1600	22	65	6	7110	7730	1
2600	0	18	9	0	2000	1300
10	1	1	5	7	0	70
0	69	-2400	68	19	0	0
1600	22	69	6	7070	7800	1
2600	0	18	9	0	2000	1300
10	1	1	5	7	0	71
0	69	-2400	68	19	0	0
1600	22	65	6	7000	7390	1
2600	0	18	9	0	2000	1300
10	1	1	5	7	0	72
0	69	-2400	68	19	0	0
1600	22	69	6	6950	7900	1
2600	0	18	9	0	2000	1300
	1	1	5	7	0	73
	0	-2400	4	20	0	0
1600	73	73	6	6900	7560	1
2600			8	0	2000	1300

200

TABLE VI (CONTINUED)

201

VI.   SIMULATED BATTLE

The following is a typical battle simulated using the STAR model.   This particular battle has a BLUE company team defending against an attacking RED tank battalion reinforced with a motorized rifle company.   A VERSATEC plot of the parametric terrain used for this battle is shown in figure 13.

BLUE Task Organization

Team C

Tank Company (-)

2 Mech Inf Platoons

2 Anti-Tank Sections

BLUE Weapon Systems Simulated:

    12 XM1 (5 per platoon, 2 in Company Headquarters)

    8 IFV (4 per Mechanized Infantry Platoon)

    4 ITV (2 per Anti-Tank Section)

    1 M113 (Forward Observer Vehicle)

Initial Ammunition Load, BLUE Weapon Systems:

    XM1:   24 rounds APDS

            16 rounds HEAT

            1000 rounds caliber 50 (stored as 100 bursts of

                    10 rounds each)

```
IFV:        6 rounds TOW
            1000 rounds Bushmaster (stored as 100 bursts of
                10 rounds each
ITV:        8 rounds TOW
DRAGON:     6 rounds per DRAGON team
M113:       FO carrier, non-firing vehicle
```

BLUE positions on the battlefield are shown in figure 5.

RED Task Organization
RED Tank Bn (+)

Team 1	Team 2	Team 3
1st Tank Co	2nd Tank Co	3rd Tank Co
1 Plt BMPs	1 Plt BMPs	1 Plt BMPs

Red Weapon Systems Simulated:

    40 T72 (13 per company team, 1 per Battalion Headquarters)

    13 BMP (4 per platoon, 1 per BMP company headquarters)

Initial Ammunition Load, RED Weapon Systems:

    T72: 22 rounds APDS
          18 rounds HEAT
          1000 rounds 12.7mm MG (stored as 100 bursts of 10
              rounds each)
    BMP: 8 rounds SAGGER
          40 rounds 73mm

Spacing of RED elements in the attack is shown in figures 3 and 4. Direction of RED attack on the battlefield is shown in figure 5. Symbols used to depict weapon systems on the battlefield are shown in figure 2.

The following pages trace the progress of the simulated battle over time. Each "time slice" of the battle is represented by a table which catalogs each round fired in the simulation, followed by a narrative description of the battle. The reader's attention is drawn to particular portions of the output which will help in understanding the progress of the battle and the workings of the model. Each narrative description is followed by a map of the battle area on which are plotted those weapon systems which were destroyed during the time period being examined. Each weapon on the map has a number adjacent to it which corresponds to the number of the round which destroyed the target.

To illustrate how one might trace the battle using the rounds fired table and the battle map, figure 1 is provided:

RND: indicates round fired (cumulative total)
  FIR: name (number) of firing weapon system
    TYPE: indicates type of weapon that fired
      AM: indicates type of ammunition fired (see Figure)
        TGT: name (number) of target weapon system
          TYPE: indicates type of weapon fired at
            TIME.V: simulated time of battle
              STAT: result of round fired
                blank-indicates miss
                DEAD-indicates target destroyed
                HIDE-indicates target in full defilade when
                  round impacted
              RANGE: range to target
                X-FIR Y-FIR: X,Y coordinate of firer
                  X-TGT Y-TGT: X,Y coordinate of target

RND	FIR	TYPE	AM	TGT	TYPE	TIME.V	STAT	RANGE	X-FIR	Y-FIR	X-TGT	Y-TGT
1	7	XM1	1	56	T72	24.6		2496	5300	5820	7238	7393
2	58	T72	1	7	XM1	31.7		2461	7213	7368	5300	5820
3	52	T72	2	28	IFV	32.9		2429	5974	7704	4760	5600
4	7	XM1	1	58	T72	32.9		2455	5300	5820	7209	7364
5	22	ITV	1	52	T72	35.0	DEAD	2799	4920	5100	5966	7696
6	27	IFV	1	56	T72	38.1	DEAD	2455	4640	5630	5935	7680
7	66	BMP	1	8	XM1	43.7	HIDE	2445	7215	7245	5200	5860

Note that the destroyed vehicle has the number 5 adjacent to it,
which corresponds to RND 5 in the rounds fired table. This row of the
table indicates that the T72 number 52 was destroyed by ITV number 22
firing ammunition type 1 (TOW) at a range of 2799 meters at 35.0 seconds
into the simulation. The X and Y coordinates of the firer and target at
the time the round impacted are also indicated.

Figure 1

```
⊟  - XM1
▆  - T72
◇  - IFV
◆  - BMP
⚲  - ITV
⚲  - DRAGON Team
⬦  - FO Vehicle
```

Figure 2   Symbols Used

Direction
of attack

BN HQ Section

(+)

(+)

(+)

150m

175m

175
m

550m

2000m
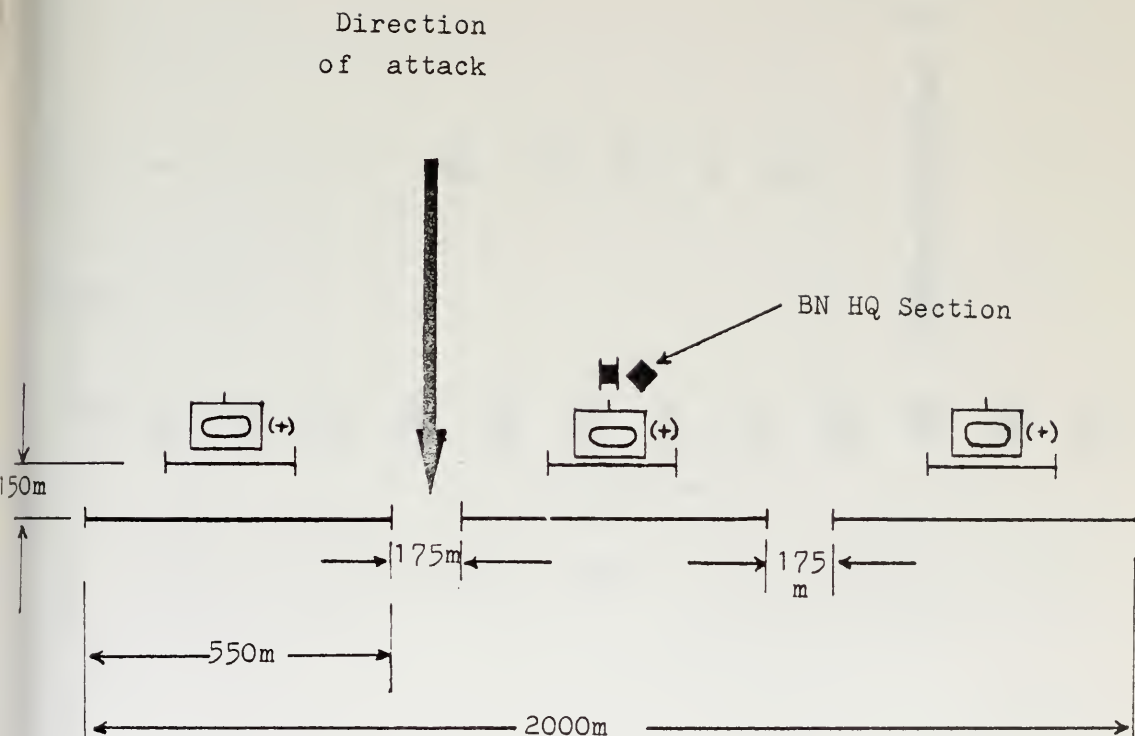
Figure 3 RED Battalion Attack Formation

207
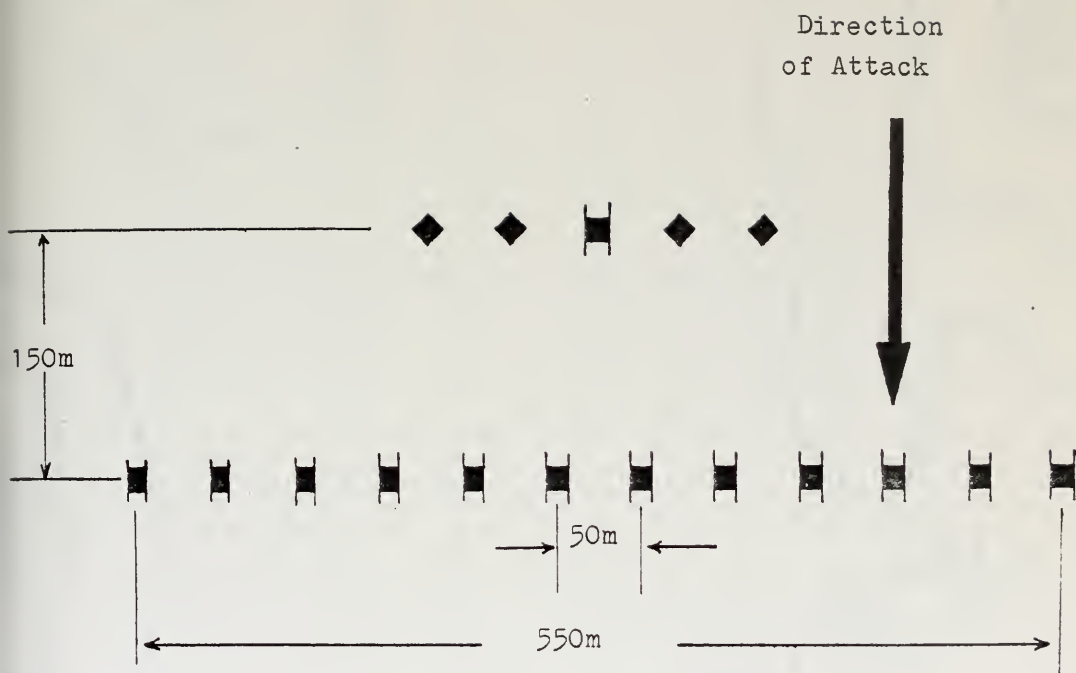
Figure 4   RED Company Formation
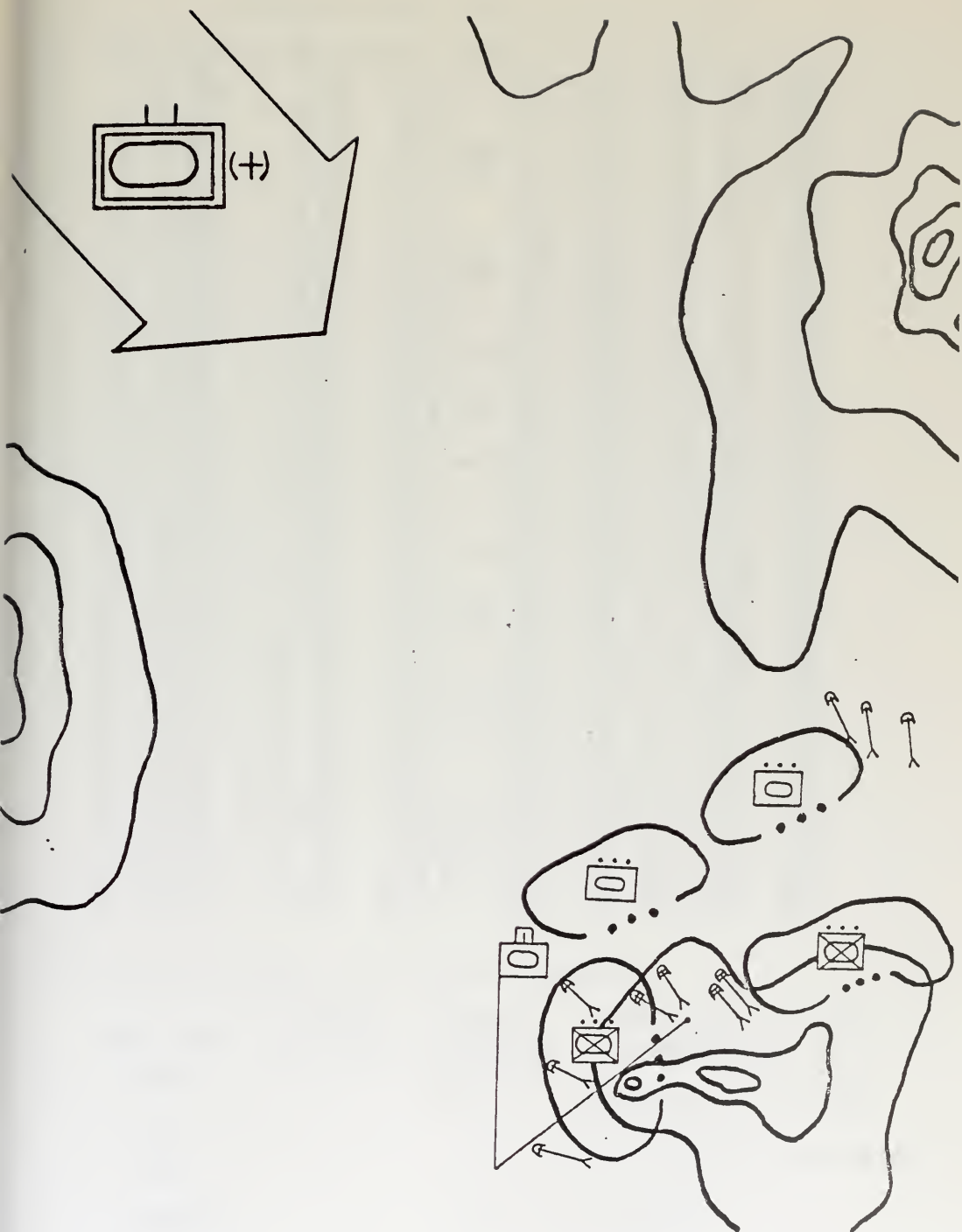
Figure 5 Initial Positions

209

Simulated Time 0 thru 104.9 seconds

RND	FIR	TYPE	AM	TGT	TYPE	TIME.V	STAT	RANGE	X-FIR	Y-FIR	X-TGT	Y-TGT
1	7	XM1	1	58	T72	24.6		2496	5300	5820	7238	7393
2	58	T72	1	7	XM1	31.7		2461	7213	7368	5300	5820
3	52	T72	2	28	IFV	32.9		2429	5974	7704	4760	5600
4	7	XM1	1	58	T72	32.9		2455	5300	5820	7209	7364
5	22	ITV	1	52	T72	35.0	DEAD	2799	4920	5100	5966	7696
6	27	IFV	1	52	T72	38.1	DEAD	2455	4640	5630	5966	7696
7	10	XM1	1	69	T72	43.7		2515	5710	5270	7575	6955
8	57	T72	1	4	XM1	44.3		2503	7133	7393	5300	5690
9	4	XM1	1	57	T72	47.7		2485	5300	5690	7121	7381
10	10	XM1	1	69	T72	50.0		2483	5710	5270	7553	6933
11	73	T72	1	10	XM1	51.0	HIDE	2530	7782	6722	5710	5270
12	65	BMP	1	9	XM1	53.3	DEAD	2869	7292	7562	5580	5260
13	6	XM1	1	58	T72	54.9		2511	5365	5500	7131	7286
14	57	T72	1	4	XM1	57.8		2435	7086	7346	5300	5690
15	4	XM1	1	57	T72	58.0		2434	5300	5690	7085	7345
16	64	BMP	1	12	XM1	59.0	DEAD	2931	7203	7603	5800	5030
17	11	XM1	1	72	T72	60.3		2515	5769	5120	7687	6737
18	6	XM1	1	58	T72	62.5		2474	5365	5500	7104	7259
19	28	IFV	1	58	T72	63.8	DEAD	2865	4760	5600	7099	7254
20	74	T72	1	11	XM1	66.8		2498	7784	6584	5760	5120
21	11	XM1	1	72	T72	67.0		2482	5760	5120	7663	6713
22	8	XM1	1	59	T72	72.6	DEAD	2346	5200	5860	7163	7143
23	7	XM1	1	69	T72	72.6		2407	5300	5820	7473	6853
24	5	XM1	1	55	T72	73.8		2501	5400	5420	6909	7414
25	72	T72	1	10	XM1	74.8		2390	7636	6686	5710	5270
26	70	T72	1	7	XM1	78.1		2428	7524	6794	5300	5820
27	69	T72	1	7	XM1	78.7	DEAD	2378	7452	6832	5300	5820
28	13	XM1	1	62	T72	79.9		2479	5850	4980	7298	6993
29	5	XM1	1	55	T72	81.6		2462	5400	5420	6882	7387
30	57	T72	1	6	XM1	86.7	DEAD	2379	6984	7244	5365	5500
31	64	BMP	1	13	XM1	87.3	DEAD	2858	7134	7534	5850	4980
32	8	XM1	2	66	BMP	87.6		2466	5200	5860	7230	7260
33	10	XM1	1	69	T72	87.9		2293	5710	5270	7419	6799
34	13	XM1	1	62	T72	88.9	DEAD	2435	5850	4980	7266	6961
35	75	T72	1	10	XM1	88.9		2390	7786	6456	5710	5270
36	27	IFV	1	46	T72	90.6	DEAD	2795	4640	5630	6680	7540
37	69	T72	1	10	XM1	92.2		2272	7404	6784	5710	5270
38	79	T72	1	10	XM1	92.6		2475	8022	6152	5710	5270
39	10	XM1	1	69	T72	93.9		2263	5710	5270	7398	6778
40	8	XM1	2	66	BMP	94.1		2445	5200	5860	7215	7245
41	77	T72	1	10	XM1	94.2	HIDE	2383	7872	6272	5710	5270
42	28	IFV	1	60	T72	97.5	DEAD	2772	4760	5600	7155	6995
43	4	XM1	1	71	T72	98.2	DEAD	2441	5300	5690	7543	6653
44	65	BMP	1	10	XM1	100.3	HIDE	2550	7123	7393	5710	5270
45	66	BMP	1	8	XM1	104.1	HIDE	2445	7215	7245	5200	5860
46	11	XM1	1	72	T72	104.9		2293	5760	5120	7529	6579

BLUE elements remaining at 104.9 seconds

Wpn Type	Number at Start	Number remaining	Average Rnds per Wpn Remaining
XM1	12	7	22 APDS, 15.71 HEAT, 1000 cal 50
IFV	8	8	5.5 TOW, 1000 Bushmaster
ITV	4	4	7.75 TOW
DRAGON	6	6	6 DRAGON

RED elements remaining at 104.9 seconds

Wpn Type	Number at Start	Number remaining	Average Rnds per Wpn Remaining
T72	40	33	21.64 APDS, 18 HEAT, 1000 12.7mm
BMP	13	13	7.61 SAGGER, 40 73mm

During the initial changes of the battle, the priority target for all weapon systems is the opposing tank, thus during this portion of the simulation those weapon systems destroyed are all tanks. Each of the BLUE tank platoons in the simulation are reduced significantly during these initial engagements (one platoon is reduced to two tanks, the second platoon is reduced to three tanks). Likewise, the red tanks are the priority target for all BLUE weapon systems.

Note that at rounds 5 and 6 T72 number 52 is impacted by two TOW rounds within 3 seconds of each other. Note also at rounds 11, 41, 44, and 45 XM1s are impacted while in full defilade indicated by the word HIDE in the STAT column.

Figure 6

Simulated time 106.3 thru 133.9 seconds

RND	FIR	TYPE	AM	TGT	TYPE	TIME.V	STAT	RANGE	X-FIR	Y-FIR	X-TGT	Y-TGT
47	82	BMP	1	8	XM1	106.3	HIDE	2728	7822	6612	5200	5860
48	67	BMP	1	2	XM1	107.4		2930	7272	7172	4970	5360
49	68	T72	1	11	XM1	109.3		2482	7834	6484	5760	5120
50	21	ITV	1	57	T72	110.4	DEAD	2802	5000	5100	6900	7160
51	11	XM1	1	72	T72	112.3		2256	5760	5120	7503	6553
52	22	ITV	1	54	T72	114.2	DEAD	2850	4920	5100	6736	7296
53	23	ITV	1	61	T72	114.9		2832	4700	5450	7134	6899
54	75	T72	1	10	XM1	115.6		2262	7691	6361	5710	5270
55	77	T72	1	4	XM1	119.0		2527	7779	6179	5300	5690
56	70	T72	1	10	XM1	119.6		2163	7378	6648	5710	5270
57	14	IFV	1	61	T72	119.7	DEAD	2849	5090	4880	7117	6882
58	64	BMP	1	1	XM1	120.5	DEAD	2859	7040	7440	4940	5500
59	27	IFV	1	44	T72	120.7	DEAD	2649	4640	5630	6493	7523
60	63	T72	1	4	XM1	120.8		2216	7233	6773	5300	5690
61	26	IFV	1	45	T72	121.5	DEAD	2781	4530	5570	6541	7491
62	53	T72	1	5	XM1	121.5		2293	6646	7346	5400	5420
63	28	IFV	1	41	T72	122.3	DEAD	2623	4760	5600	6348	7688
64	5	XM1	1	72	T72	126.6	DEAD	2320	5400	5420	7452	6502
65	69	T72	1	4	XM1	128.8	DEAD	2198	7275	6655	5300	5690
66	65	BMP	1	3	ITV	129.4	DEAD	2820	7049	7319	4910	5480
67	10	XM1	1	69	T72	131.6	DEAD	2075	5710	5270	7265	6645
68	56	T72	1	2	XM1	133.2		2530	6749	7159	4970	5360
69	24	ITV	1	42	T72	133.8	DEAD	2759	4650	5430	6357	7597
70	73	T72	1	10	XM1	133.9		2120	7486	6426	5710	5270

BLUE Elements Remaining at 133.9 seconds

Wpn Type	Number at Start	Number remaining	Average Rnds per Wpn remaining
XM1	12	5	21 APDS, 15.6 HEAT, 1000 cal 50
IFV	8	8	5 TOW, 1000 Bushmaster
ITV	4	4	6.75 TOW
DRAGON	6	6	6 DRAGON

RED Elements Remaining at 133.9 seconds

Wpn Type	Number at Start	Number remaining	Average Rnds per Wpn remaining
T72	40	24	21.42 APDS, 18 HEAT, 1000 12.7mm
BMP	13	13	7.31 SAGGER, 40 73mm

Priority target for all weapon systems continues to be the opposing tank forces. Each of the two BLUE tank platoons is reduced to two tanks. In addition, one of the headquarters tanks of the defending team is destroyed.

Note that at rounds 66, ITV number 3 was destroyed. This vehicle is listed as an ITV for lethality purposes, but in fact, is the FO vehicle.

Note also that during the first 133 seconds of the simulation the attacking tank force is reduced to 60% of its original strength.

Figure 7

215
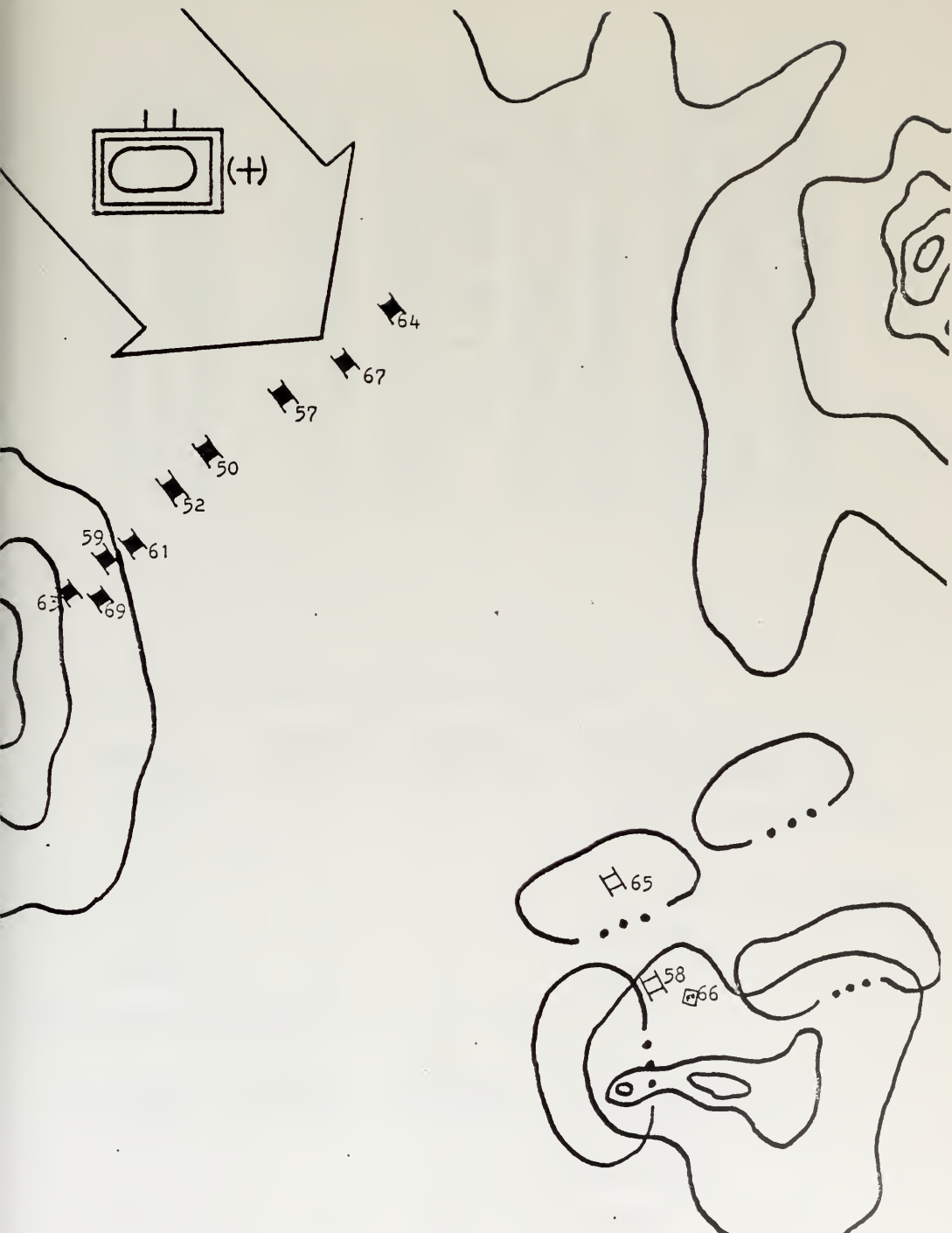
Simulated time 135.4 thru 158.7 seconds

RND	FIR	TYPE	AM	TGT	TYPE	TIME.V	STAT	RANGE	X-FIR	Y-FIR	X-TGT	Y-TGT
71	79	T72	1	10	XM1	135.4		2282	7871	6001	5710	5270
72	67	BMP	1	2	XM1	135.6	DEAD	2832	7203	7103	4970	5360
73	15	IFV	1	69	T72	136.1	DEAD	2819	5150	4780	7265	6645
74	63	T72	1	10	XM1	137.0		2059	7176	6716	5710	5270
75	66	BMP	1	27	IFV	138.0	DEAD	2908	7120	7150	4640	5630
76	77	T72	1	10	XM1	138.2	DEAD	2171	7711	6111	5710	5270
77	21	ITV	1	55	T72	138.4	DEAD	2678	5000	5100	6681	7186
78	53	T72	1	5	XM1	140.1	DEAD	2202	6570	7280	5400	5420
79	5	XM1	1	56	T72	140.2		2166	5400	5420	6724	7134
80	81	BMP	1	5	XM1	141.9	DEAD	2508	7643	6543	5400	5420
81	22	ITV	1	53	T72	143.0	DEAD	2725	4920	5100	6570	7270
82	8	XM1	2	67	BMP	144.9		2294	5200	5860	7157	7057
83	14	IFV	1	70	T72	150.7	DEAD	2736	5099	4880	7267	6537
84	8	XM1	2	67	BMP	153.2		2237	5200	5860	7115	7015
85	26	IFV	1	43	T72	154.1	DEAD	2624	4530	5570	6355	7455
86	32	T72	1	11	XM1	154.6		2514	7073	7263	5760	5120
87	28	IFV	1	63	T72	155.4	DEAD	2575	4760	5600	7110	6650
88	11	XM1	1	76	T72	156.0		2107	5760	5120	7599	6149
89	68	T72	1	11	XM1	157.8		2245	7662	6312	5760	5120
90	51	T72	1	11	XM1	158.3		2447	6890	7290	5760	5120
91	82	BMP	1	11	XM1	158.7		2263	7619	6409	5760	5120

BLUE Elements Remaining at 158.7 seconds

Wpn Type	Number at Start	Number remaining	Average Rnds per Wpn remaining
XM1	12	2	21 APDS, 14 HEAT, 1000 cal 50
IFV	8	7	4.71 TOW, 1000 Bushmaster
ITV	4	4	6.25 TOW
DRAGON	6	6	6 DRAGON

RED Elements Remaining at 158.7 seconds

Wpn Type	Number at Start	Number remaining	Average Rnds per Wpn remaining
T72	40	19	20.21 APDS, 18 HEAT, 1000 cal 50
BMP	13	13	7 SAGGER, 40 73mm

216

Red tank systems continue to be the priority target for BLUE systems and continue to be attritted significantly. To this point in the battle no BMPs have been eliminated from the attacking force. Note that T72 number 69 (round 73) was previously destroyed at round 67. Round 73 is the impact of a TOW round fired by an IFV. The impact of this second round on T72 number 69 is largely a consequence of the long time of flight of the TOW round which was actually fired prior to the impact of round 67.

The BLUE defenders have been reduced to only two tanks, one in each of the tank platoons. The reduction in the number of defending tanks should cause IFV and ITV systems to become the priority target for the attackers.
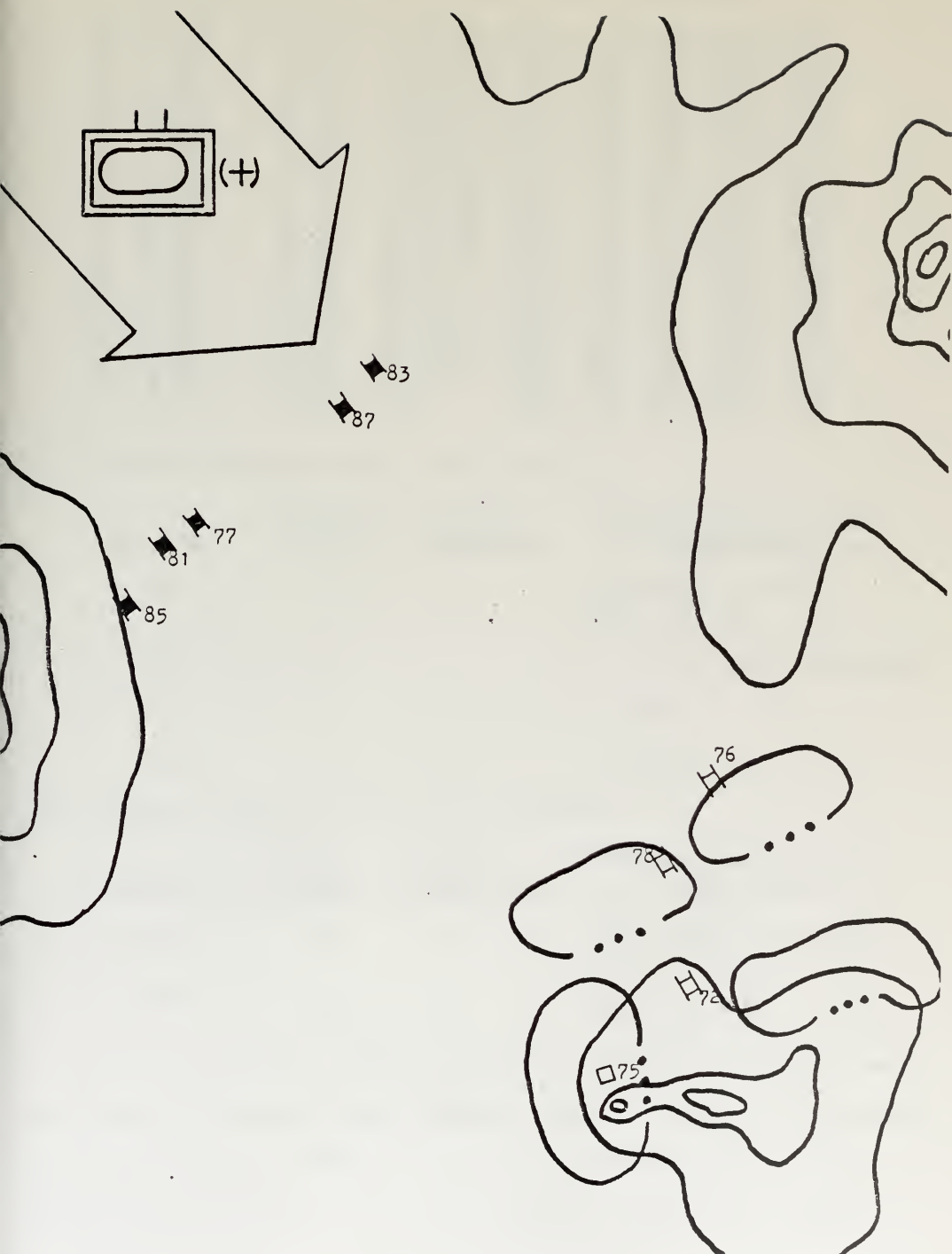
Figure 8

218

Simulated time 161.7 thru 201.9 seconds

RND	FIR	TYPE	AM	TGT	TYPE	TIME.V	STAT	RANGE	X-FIR	Y-FIR	X-TGT	Y-TGT
92	23	ITV	1	39	T72	161.7	DEAD	2630	4700	5450	6048	7708
93	11	XM1	1	76	T72	162.8	DEAD	2074	5760	5120	7574	6124
94	65	BMP	1	21	ITV	163.3	DEAD	2883	6952	7222	5000	5100
95	66	BMP	1	28	IFV	163.9	DEAD	2739	7059	7089	4760	5600
96	16	IFV	1	75	T72	164.2	DEAD	2835	5120	4680	7520	6190
97	64	BMP	1	11	XM1	165.0	HIDE	2434	6880	7280	5760	5120
98	21	ITV	1	63	T72	165.6	DEAD	2619	5000	5100	7110	6650
99	15	IFV	1	73	T72	166.6	DEAD	2697	5150	4780	7371	6311
100	67	BMP	1	23	ITV	179.6	DEAD	2794	7054	6954	4700	5450
101	26	IFV	1	40	T72	182.6	DEAD	2502	4530	5570	6054	7554
102	22	ITV	1	56	T72	186.2	DEAD	2489	4920	5100	6562	6972
103	82	BMP	1	22	ITV	187.9	DEAD	2903	7546	6336	4920	5100
104	16	IFV	1	77	T72	189.5	DEAD	2715	5120	4680	7530	5930
105	8	XM1	1	68	T72	189.8		2373	5200	5860	7549	6199
106	14	IFV	1	74	T72	193.7	DEAD	2572	5090	4880	7335	6135
107	50	BMP	1	22	ITV	194.7	DEAD	2838	6242	7612	4920	5100
108	66	BMP	1	26	IFV	194.8	DEAD	2837	6976	7006	4530	5570
109	65	BMP	1	22	ITV	196.1	DEAD	2807	6859	7129	4920	5100
110	8	XM1	1	68	T72	197.5		2342	5200	5860	7522	6172
111	11	XM1	1	32	T72	201.9		2285	5760	5120	6906	7096

BLUE elements remaining at 201.9 seconds

Wpn Type	Number at Start	Number remaining	Average Rnds per Wpn remaining
XM1	12	2	19 APDS, 14 HEAT, 1000 cal 50
IFV	8	5	4.4 TOW, 1000 Bushmaster
ITV	4	1	7 TOW
DRAGON	6	6	6 DRAGON

RED elements remaining at 201.9 seconds

Wpn Type	Number at Start	Number remaining	Average Rnds per Wpn remaining
T72	40	11	21.45 APDS, 18 HEAT, 1000 12.7mm
BMP	13	13	6.38 SAGGER, 40 73mm

At this point in the simulation, BLUE long range systems have been attrited significantly. Each of the tank platoons has one remaining vehicle. One of the Mech platoons has been reduced to only 1 IFV and 3 DRAGON teams, and there

remains only one operational ITV in the simulation. The
battle has become virtually a SAGGER versus TOW duel.

BLUE weapons continue to target attacking tanks, while
the scarcity of BLUE tanks make the IFV and ITV priority
targets for the attackers.
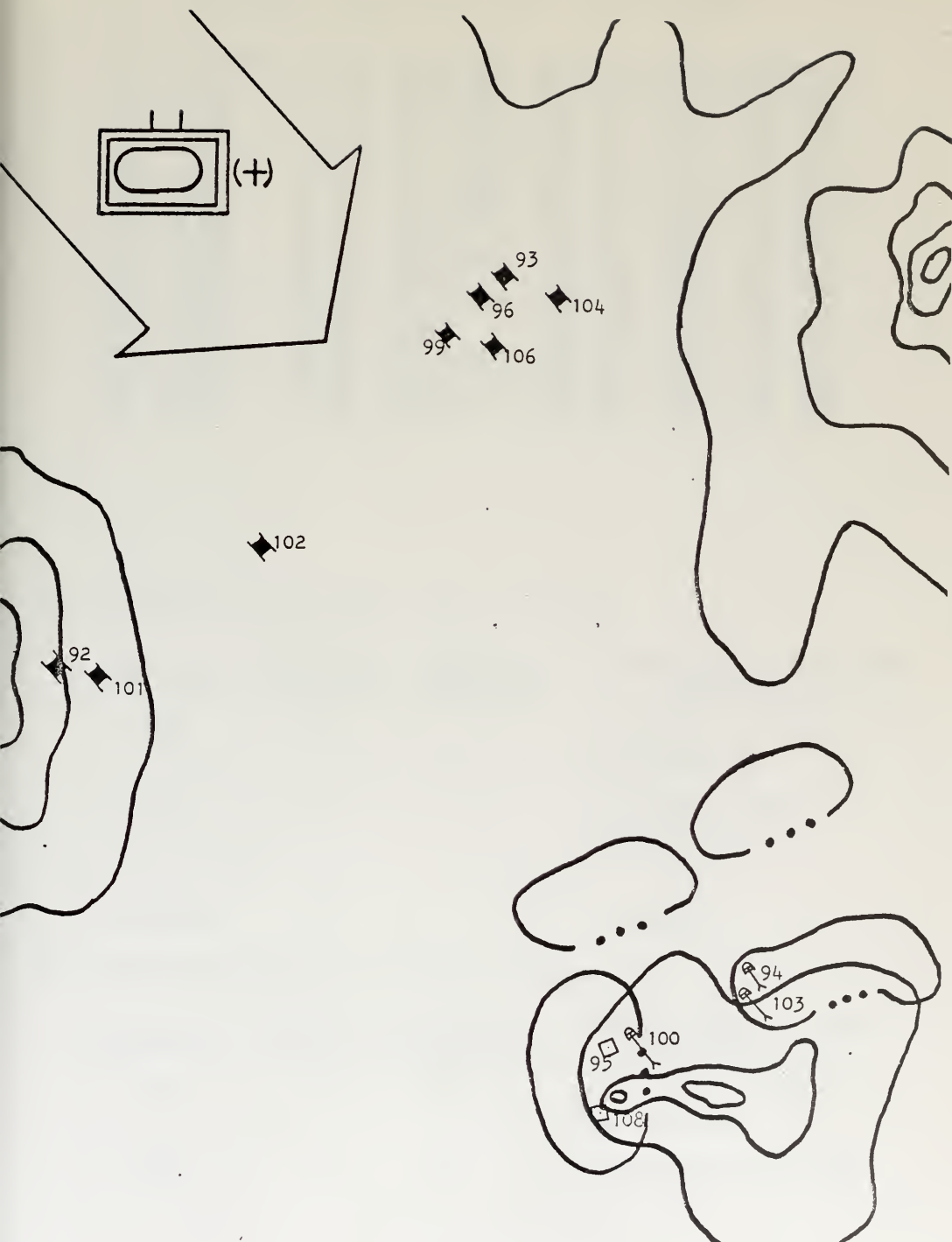
Figure 9

Simulated time 203.8 thru 272.8 seconds

RND	FIR	TYPE	AM	TGT	TYPE	TIME.V	STAT	RANGE	X-FIR	Y-FIR	X-TGT	Y-TGT
112	51	T72	1	11	XM1	203.8	DEAD	2231	6730	7130	5760	5120
113	24	ITV	1	32	T72	209.8	DEAD	2766	4650	5430	6878	7068
114	67	BMP	1	24	ITV	210.3	DEAD	2730	6969	6869	4650	5430
115	81	BMP	1	24	ITV	210.7	DEAD	2850	7372	6272	4650	5430
116	82	BMP	1	14	IFV	215.0	DEAD	2766	7481	6271	5090	4880
117	17	IFV	1	68	T72	215.3	DEAD	2833	5100	4540	7459	6109
118	15	IFV	1	51	T72	215.8	DEAD	2772	5150	4780	6687	7087
119	16	IFV	1	81	BMP	219.5	DEAD	2698	5120	4680	7329	6229
120	49	BMP	1	14	IFV	219.8	DEAD	2916	6075	7625	5090	4880
121	14	IFV	1	78	T72	222.0	.	2563	5090	4880	7485	5765
122	65	BMP	1	14	IFV	223.4	DEAD	2768	6793	7063	5090	4880
123	8	XM1	2	67	BMP	234.2		1876	5200	5860	6851	6751
124	82	BMP	.1	16	IFV	241.1	DEAD	2762	7420	6210	5120	4680
125	78	T72	1	8	XM1	242.4		2229	7423	5693	5200	5860
126	8	XM1	2	67	BMP	245.6		1823	5200	5860	6812	6712
127	15	IFV	1	84	BMP	246.3	DEAD	2555	5150	4780	7479	5829
128	16	IFV	1	82	BMP	248.4		2712	5120	4680	7384	6174
129	67	BMP	1	8	XM1	251.3	HIDE	1823	6812	6712	5200	5860
130	82	BMP	1	15	IFV	272.8		2554	7326	6116	5150	4780

BLUE elements remaining at 272.8 seconds

Wpn Type	Number at Start	Number remaining	Average Rnds per Wpn remaining
XM1	12	1	21 APDS, 10 HEAT, 1000 cal 50
IFV	8	3	3.33 TOW, 1000 Bushmaster
ITV	4	0	-------
DRAGON	6	6	6 DRAGON

RED elements remaining at 272.8 seconds

Wpn Type	Number at Start	Number remaining	Average Rnds per Wpn remaining
T72	40	8	21.63 APDS, 18 HEAT, 1000 12.7mm
BMP	13	11	5.54 SAGGER, 40 73mm

At this point in the simulation, the BLUE force has been reduced to 1 tank, 3 IFVs, and 6 DRAGON teams. The RED tank force has been reduced significantly, but the attacking BMP force remains essentially intact.

Available TOW ammunition has been reduced to only 10 rounds available to the defending force. IFVs number 15 and 17 have two rounds each, while IFV number 25 has 6 TOWs remaining. All defending ITVs have been destroyed.
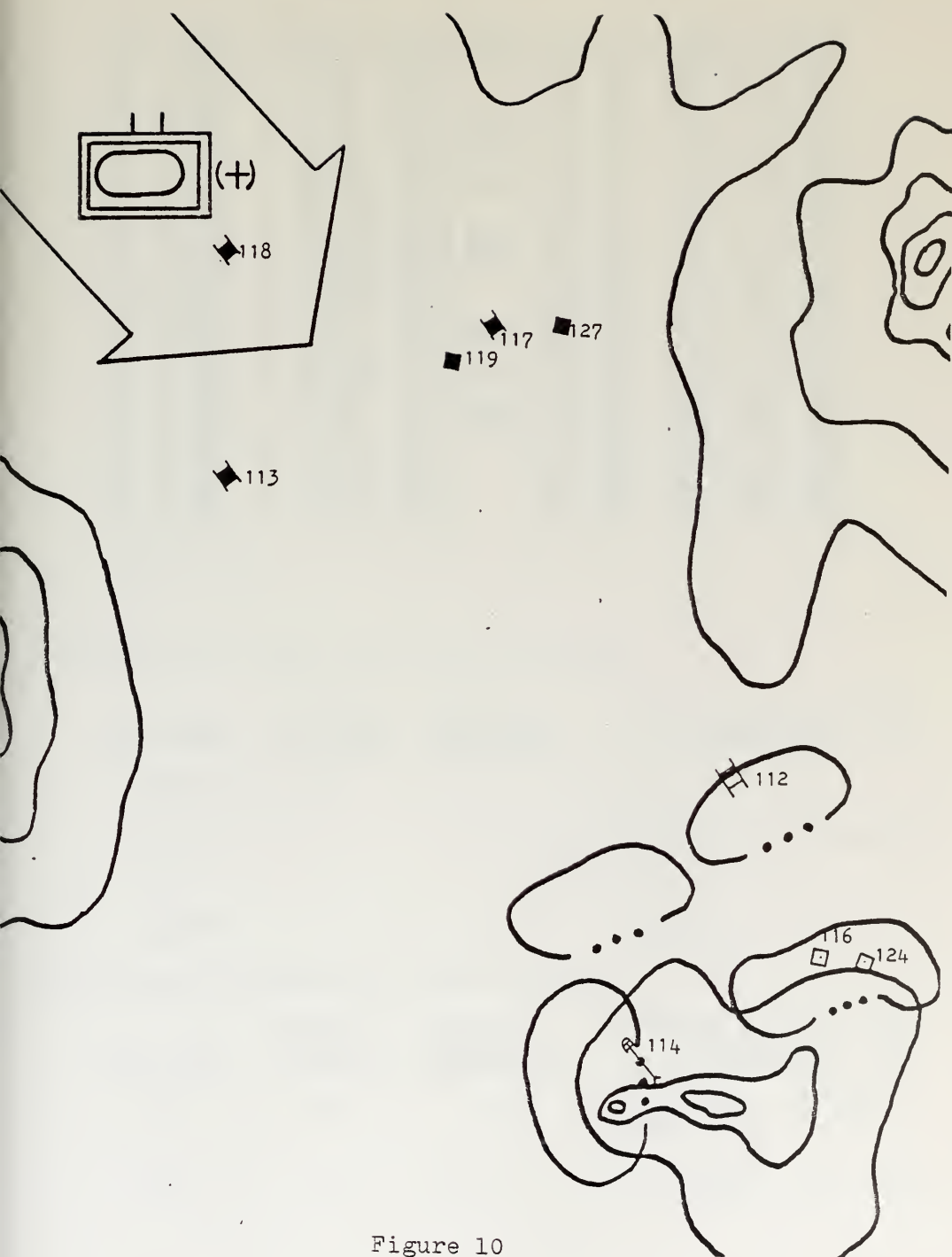
Figure 10

Simulated time 277.5 thru 399.6 seconds

RND	FIR	TYPE	AM	TGT	TYPE	TIME.V	STAT	RANGE	X-FIR	Y-FIR	X-TGT	Y-TGT
131	15	IFV	1	65	BMP	277.5	DEAD	2304	5150	4780	6567	6597
132	8	XM1	1	35	T72	285.7		1732	5200	5860	5290	7590
133	35	T72	1	8	XM1	285.8		1732	5290	7590	5200	5860
134	8	XM1	1	35	T72	292.7		1706	5200	5860	5265	7565
135	83	BMP	1	8	XM1	298.3	HIDE	2083	7282	5782	5200	5860
136	82	BMP	1	15	IFV	298.4	DEAD	2463	7260	6050	5150	4780
137	15	IFV	1	67	BMP	308.5		2149	5150	4780	6528	6428
138	50	BMP	3	30	DRAG	312.8	DEAD	1713	5657	7027	4550	5720
139	64	BMP	3	18	DRAG	313.1		1745	6147	6547	5840	4830
140	67	BMP	3	31	DRAG	315.5		1906	6494	6394	4700	5750
141	36	T72	3	31	DRAG	316.5	DEAD	1750	5281	7401	4700	5750
142	48	BMP	3	31	DRAG	316.7	DEAD	1834	5540	7380	4700	5750
143	33	BMP	3	18	DRAG	327.5		1897	6542	6592	5840	4830
144	64	BMP	3	18	DRAG	331.9		1638	6054	6454	5840	4830
145	65	BMP	3	19	DRAG	332.6		1841	6252	6522	5790	4740
146	83	BMP	1	8	XM1	343.8	DEAD	1922	7105	5605	5200	5860
147	65	BMP	3	19	DRAG	344.3		1771	6194	6464	5790	4740
148	67	BMP	3	20	DRAG	345.8	DEAD	1703	6344	6244	5670	4680
149	64	BMP	3	18	DRAG	346.6		1558	5981	6381	5840	4830
150	33	BMP	1	8	XM1	350.0	DEAD	1456	6488	6538	5200	5860
151	65	BMP	3	19	DRAG	364.6	DEAD	1652	6084	6364	5790	4740
152	64	BMP	3	18	DRAG	366.8		1452	5881	6281	5840	4830
153	33	BMP	3	18	DRAG	369.9		1701	6390	6440	5840	4830
154	64	BMP	3	18	DRAG	387.2		1352	5780	6180	5840	4830
155	33	BMP	3	18	DRAG	390.3		1574	6289	6339	5840	4830
156	64	BMP	3	18	DRAG	399.6		1295	5719	6119	5840	4830

BLUE elements remaining at 399.6 seconds

Wpn Type	Number at Start	Number remaining	Average Rnds per Wpn remaining
XM1	12	0	---------------
IFV	8	2	4 TOW, 1000 Bushmaster
ITV	4	0	---------------
DRAGON	6	2	6 DRAGON

RED elements remaining at 399.6 seconds

Wpn Type	Number at Start	Number remaining	Average Rnds per Wpn remaining
T72	40	8	21.5 APDS, 18 HEAT, 998 12.7mm
BMP	13	10	5.3 SAGGER, 38.4 73mm

At this point in the simulation, the BLUE tank force and ITVs have been eliminated. Each of the Mech platoons has one IFV and one DRAGON team remaining. The lack of BLUE weapon systems remaining in the simulation has caused the DRAGON teams to assume highest priority in the target lists of some of the attackers. The DRAGONS are being engaged at ranges beyond 1000 meters and thus are unable to return fire. Note that DRAGONS are being engaged by ammunition type 3 (73mm for BMPs and 12.7mm for T72s).

Note also at round 136 that IFV number 15 is destroyed, but at this time he has a TOW round in flight which impacts at round 137. The model takes into account the fact that a wire guided round is "unguided" when its carrier is destroyed, and although the round impacts it will always be assessed as a miss.
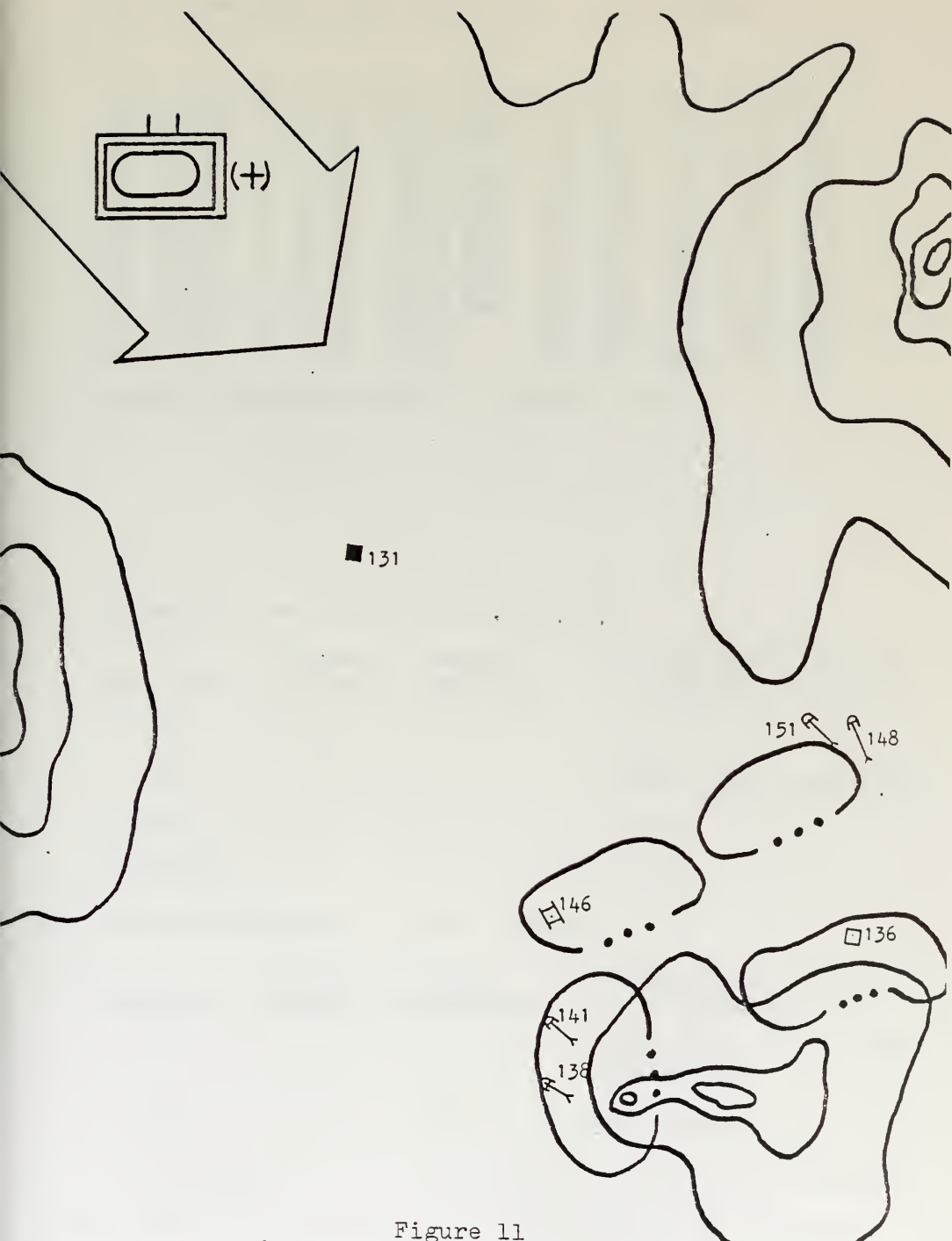
131

151 148

146

136

141

138

Figure 11

227

RND	FIR	TYPE	AM	TGT	TYPE	TIME.V	STAT	RANGE	X-FIR	Y-FIR	X-TGT	Y-TGT
157	33	BMP	3	18	DRAG	402.3	DEAD	1501	6230	6280	5840	4830
158	17	IFV	1	78	T72	490.8	DEAD	1470	5100	4540	6545	4815
159	80	T72	2	17	IFV	498.1		1579	6676	4626	5100	4540
160	79	T72	2	17	IFV	498.7	DEAD	1505	6594	4724	5100	4540
161	29	DRAG	2	36	T72	563.8	DEAD	887	4430	5640	4407	6527
162	29	DRAG	2	37	T72	578.8	DEAD	754	4430	5640	4394	6394
163	25	IFV	3	49	BMP	586.3	DEAD	292	4370	5540	4260	5810
164	25	IFV	3	50	BMP	612.3	DEAD	195	4370	5540	4175	5545
165	25	IFV	1	35	T72	631.8	DEAD	880	4370	5540	4066	6366
166	25	IFV	1	38	T72	650.6	DEAD	524	4370	5540	4250	6050
167	25	IFV	3	48	BMP	686.7	DEAD	550	4370	5540	4232	6072
168	25	IFV	3	47	BMP	705.2	·	616	4370	5540	4087	6087
169	25	IFV	3	47	BMP	721.7		596	4370	5540	4028	6028
170	25	IFV	3	47	BMP	738.4		587	4370	5540	3969	5969

SIMULATION STOPPED AT 750.000   LOS CALLS = 4696

.

BLUE elements remaining at 750.0 seconds

Wpn Type	Number at Start	Number remaining	Average Rnds per Wpn remaining
XM1	12	0	--------------
IFV	8	1	4 TOW, 940 Bushmaster
ITV	4	0	--------------
DRAGON	6	1	4 DRAGON

RED elements remaining at 750.0 seconds

Wpn Type	Number at Start	Number remaining	Average Rnds per Wpn remaining
T72	40	3	21.33 APDS, 17.33 HEAT, 1000 12.7mm
BMP	13	7	4.43 SAGGER, 37.85 73mm

At termination of the simulation, one IFV and one DRAGON team remain of the BLUE force, while three T72s and seven BMPs remain of the RED force.

The attacking force had closed to a range where the DRAGON was able to bring effective fire on the RED force. Notice that at these reduced ranges the IFV continues to engage T72s with ammunition type 1 (TOW), but selects ammunition type 3 (Bushmaster) to fire at the thinner skinned BMPs.

The attacking force remains oriented toward its direction of movement, and thus never "sees" the remaining defenders as they pass the BLUE positions. This enables the defenders to destroy several RED vehicles during the last few minutes of the simulation without drawing any significant answering fires.

Figure 12

230
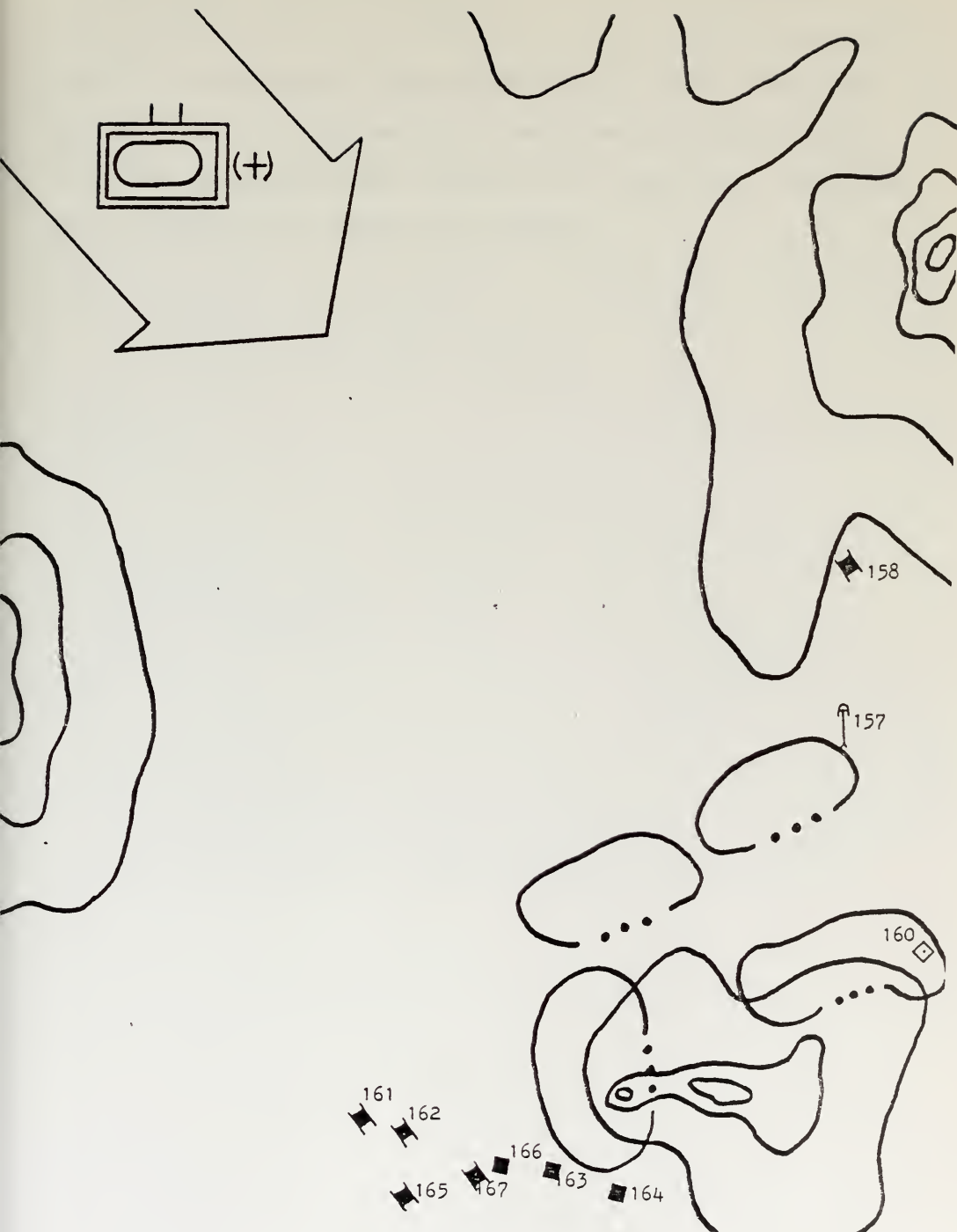
The reader is cautioned not to draw any analytical conclusions from the preceding simulation. This simulation was run to demonstrate the capabilities of the STAR model. The data used for the exercise was greatly simplified and in no way represents what might be the results of the model run in support of a particular study.
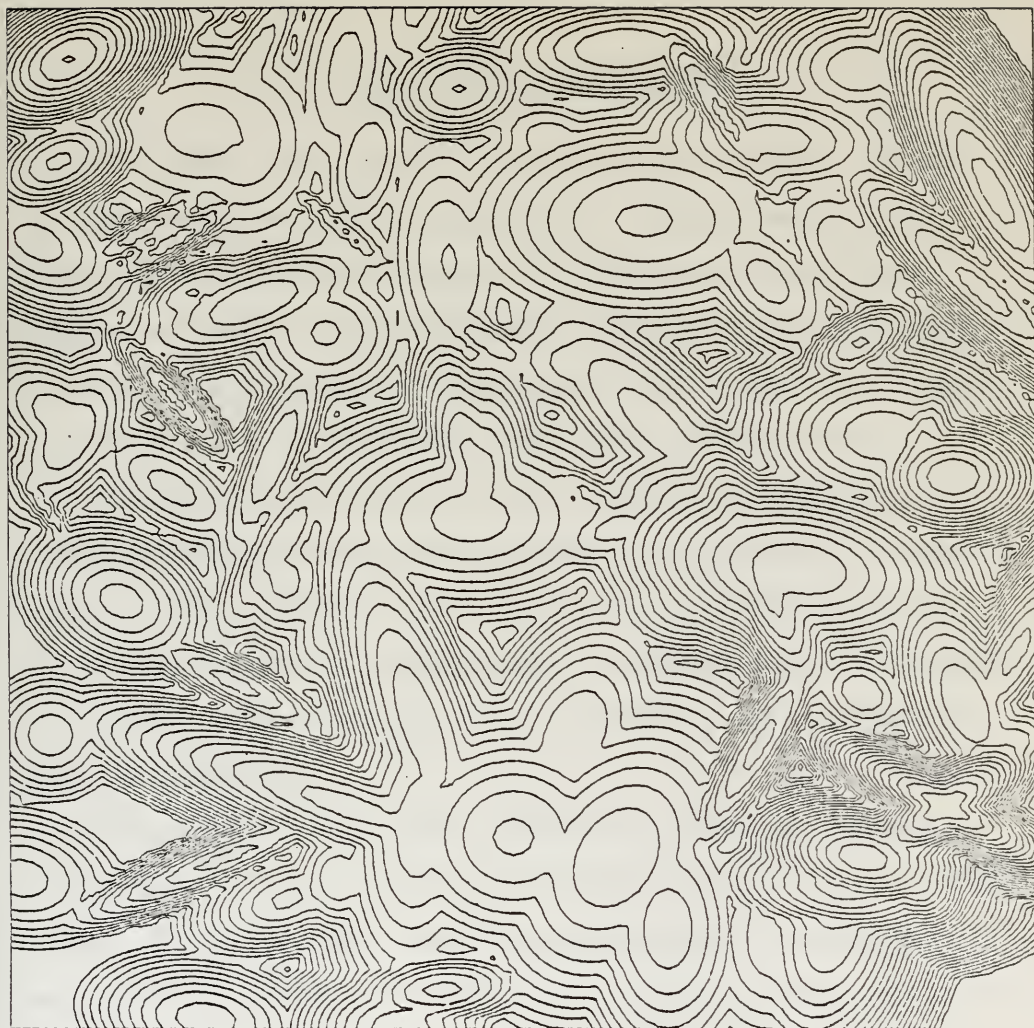
Versatec Plot of Parametric Terrain

Figure 13

VII. FUTURE MODEL ENHANCEMENT

The model described in the preceding chapters should not be viewed as the final product. Enhancement of the existing model is the subject of several masters degree theses currently under development at the Naval Postgraduate School. The purpose of this chapter is to give the reader a general idea of projected additions to the model.

As mentioned in Chapter I, a close air support/air defense module is under development for inclusion in the model. The availability of this module is projected to be mid-April 1979. In addition, a working artillery module is now available, and will be integrated into the model by mid-December 1978.

The criteria for target selection using the danger state arrays is not considered sufficient for future model uses. It is possible that the interaction of such factors as ammunition remaining, whether an enemy element has fired, anticipated time of ammunition resupply, etc., could alter the priorities calculated from the danger state arrays. Research is currently being conducted to determine what additional factors are important in target selection and how these factors should be included in target selection logic.

Ammunition and fuel redistribution and resupply are considered vital to the success of any military operation.

Future versions of the model will include resupply of fuel and ammunition to include the introduction of resupply vehicles to the simulation to allow dynamic resupply to be played.

A Battalion communication model is currently under development which will incorporate electronic warfare considerations. This model will contain a realistic propagation model, incomplete transmission detection and retransmission, frequency change effects, play of message traffic and play of RED intercept and jamming.

Dynamic route selection is being considered for future model inclusion. A module which considers enemy locations, the terrain, and the tactical situation, and dynamically computes routes for elements has been proposed to enhance the existing model. This module would be a user specified option and would allow pre-selection of routes or dynamic selection of routes as desired by the user.

It is difficult to predict when all of the above mentioned modules will be available for integration into the STAR model. It is intended that enhancement of the model be a continuing effort at the Naval Postgraduate School, the objective of which is to provide a versatile mid-resolution combat model for a wide variety of military applications.

APPENDIX A

FLOWCHART

This appendix presents the event sequence a single XMl
would follow through the model.

Within the flowchart the following non-standard symbol
is used:

an event

Explanatory text accompanies the flowchart. For
details concerning the processing within an event, see
Chapter IV.
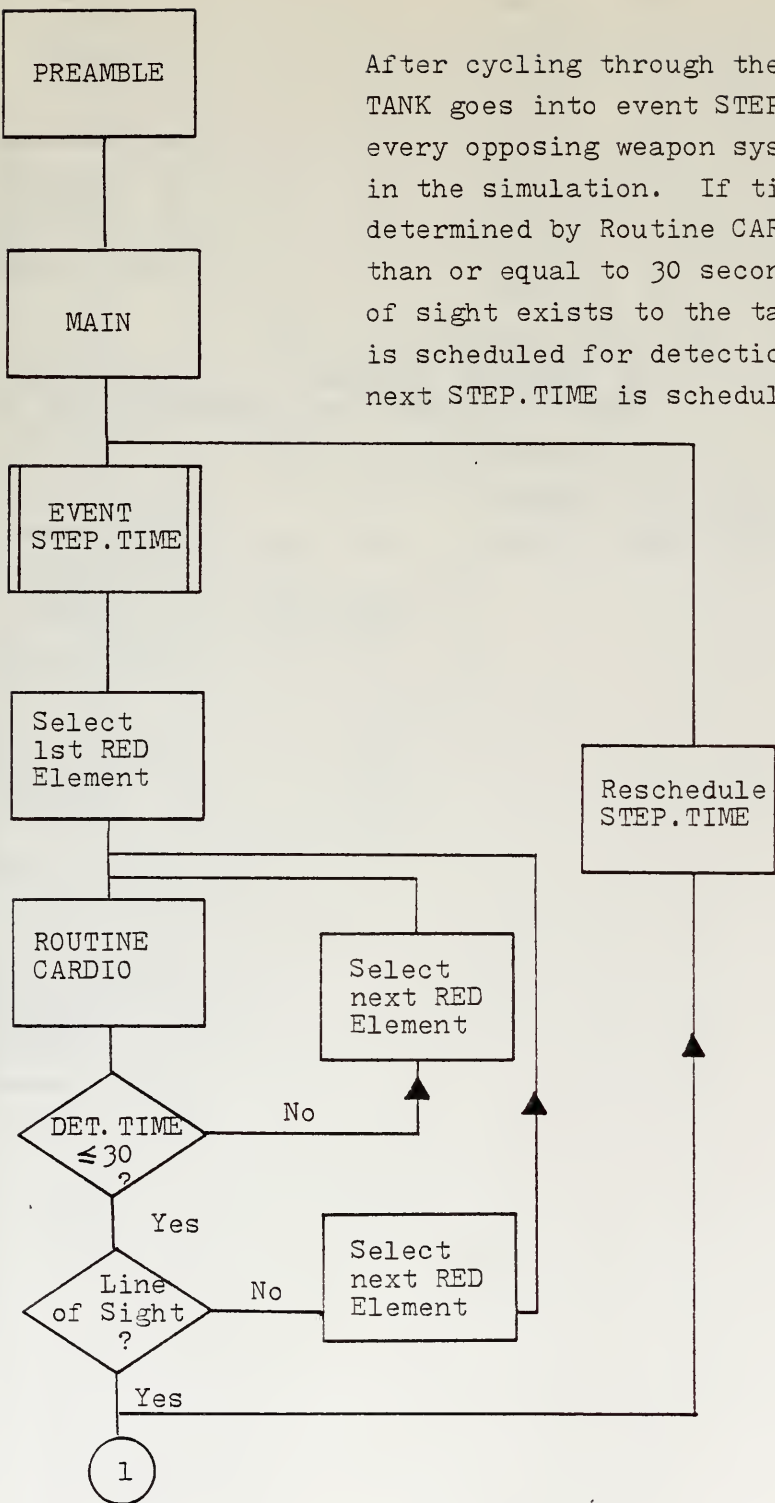
```
┌─────────────┐
│             │              After cycling through the PREAMBLE and MAIN,
│  PREAMBLE   │              TANK goes into event STEP.TIME and looks at
│             │              every opposing weapon system still active
└─────────────┘              in the simulation.  If time to detect
       │                     determined by Routine CARDIO is less
       │                     than or equal to 30 seconds, and line
┌─────────────┐              of sight exists to the target, the target
│             │              is scheduled for detection.  In any case
│    MAIN     │              next STEP.TIME is scheduled.
│             │
└─────────────┘
       │
┌─────────────┐
║             ║
║   EVENT     ║
║ STEP.TIME   ║
║             ║
└─────────────┘
       │
┌─────────────┐
│  Select     │                              ┌──────────────┐
│  1st RED    │                              │              │
│  Element    │                              │  Reschedule  │
└─────────────┘                              │  STEP.TIME   │
       │                                     │              │
┌─────────────┐        ┌──────────────┐      └──────────────┘
│  ROUTINE    │        │  Select      │
│  CARDIO     │        │  next RED    │
│             │        │  Element     │
└─────────────┘        └──────────────┘
       │
      ╱ ╲
    ╱ DET. ╲     No
   ╱ TIME    ╲─────────►
   ╲  ≤ 30   ╱
    ╲   ?   ╱
      ╲ ╱
       │ Yes
      ╱ ╲            ┌──────────────┐
    ╱ Line ╲   No    │  Select      │
   ╱  of    ╲────────│  next RED    │
   ╲ Sight  ╱        │  Element     │
    ╲  ?   ╱         └──────────────┘
      ╲ ╱
       │ Yes
      ╱─╲
     │ 1 │
      ╲─╱
```

236

```
        ( 3 )
          |
    ┌─────────────┐
    ║ Event       ║
    ║ IMPACT      ║
    └─────────────┘
          |
          |
      ╱Target╲      No      ┌─────────────┐
     ╱ killed  ╲────────────│ Routine     │
     ╲    ?    ╱            │ WE..MISS    │
      ╲      ╱              └─────────────┘
        | Yes                     |
  ┌─────────────┐           ┌─────────────┐
  │ Routine     │           │ Reschedule  │
  │ TALLY.HIT   │           │ FIRE        │
  │ STATE       │           └─────────────┘
  └─────────────┘                 |
          |                     ( B )
  ┌─────────────┐
  │ Routine     │
  │ WE.HIT      │
  └─────────────┘
          |
  ┌─────────────┐
  │ Reschedule  │
  │ TARGET.     │
  │ SELECT      │
  └─────────────┘
          |
        ( A )
```
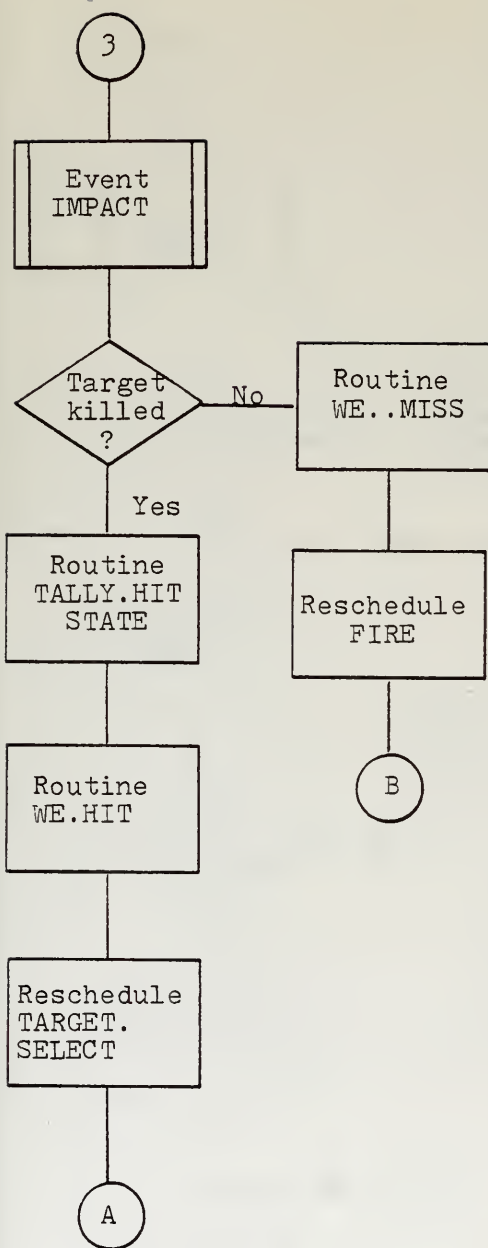
Routine IMPACT occurs at the time specified in event FIRE based on the time of flight of the projectile. Determination is made as to whether the round hits or misses the target, and the proper routine is called to tally the result of the round. If the round was a miss, the target is engaged again by rescheduling event FIRE. If the round is assessed as a hit, the firer selects a new target by scheduling the event TARGET.SELECT.

```
        ( 1 )
          |
    +-----------+
    |  Event    |
    |  DETECT   |
    +-----------+
          |
        /   \                    +-------------+
       / Line \      No          | Go to next  |
      < of Sight >--------------->| Scheduled   |
       \   ?   /                  | Event       |
        \   /                     +-------------+
          | Yes
    +-----------+
    | Routine   |<---------------------+
    | PROXIMITY.|                      |
    | DETECT    |                +-------------+
    +-----------+                | Check next  |
          |                      | tank in     |
        /   \        No          | platoon     |
       / Tank \----------------->+-------------+
      < within  >                      ^
       \75meters/                      |
        \   /                          |
          | Yes                        |
    +-----------+                      |
    | Routine   |----------------------+
    | LIST.UPDATE
    +-----------+
          |
          |<-------( A )
    +-----------+
    | TARGET.   |
    | SELECT    |
    +-----------+
          |
        ( 2 )
```
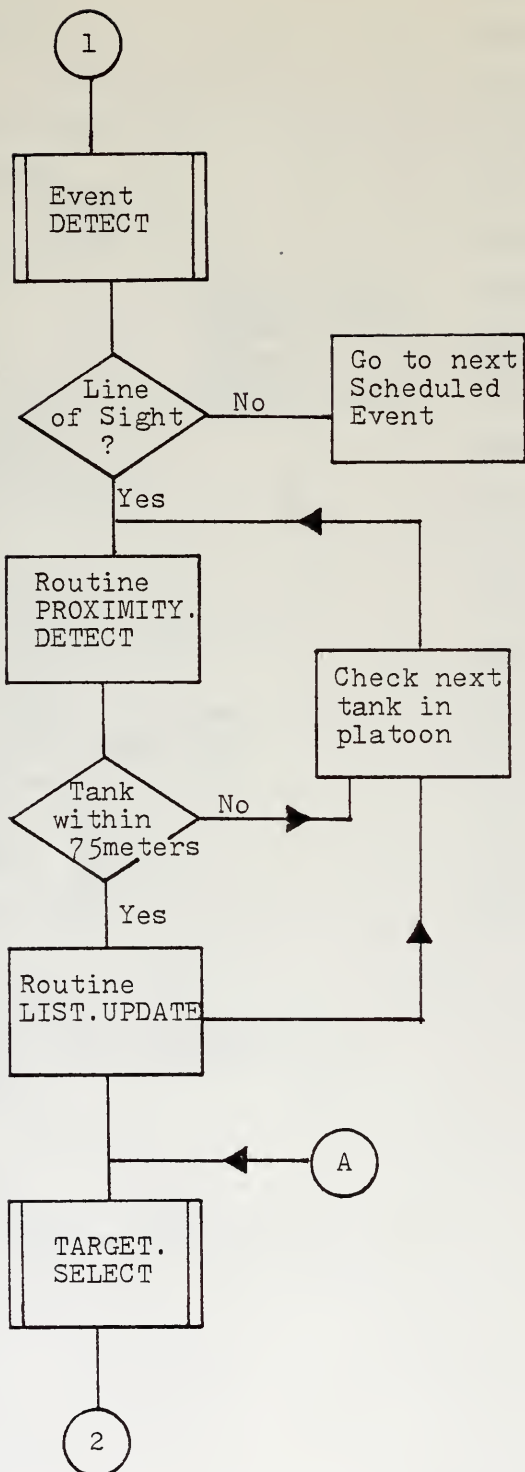
Event DETECT takes place at time scheduled in event STEP.TIME. Event DETECT first checks to see if line of sight exists between firer and target; if so, routine PROXIMITY. DETECT is called which checks all weapon systems in the detected vehicle's platoon. Routine LIST. UPDATE is called for the detected vehicle and any proximity detections. LIST.UPDATE places targets in the firer's list of currently detected targets and schedules event TARGET.SELECT.

238
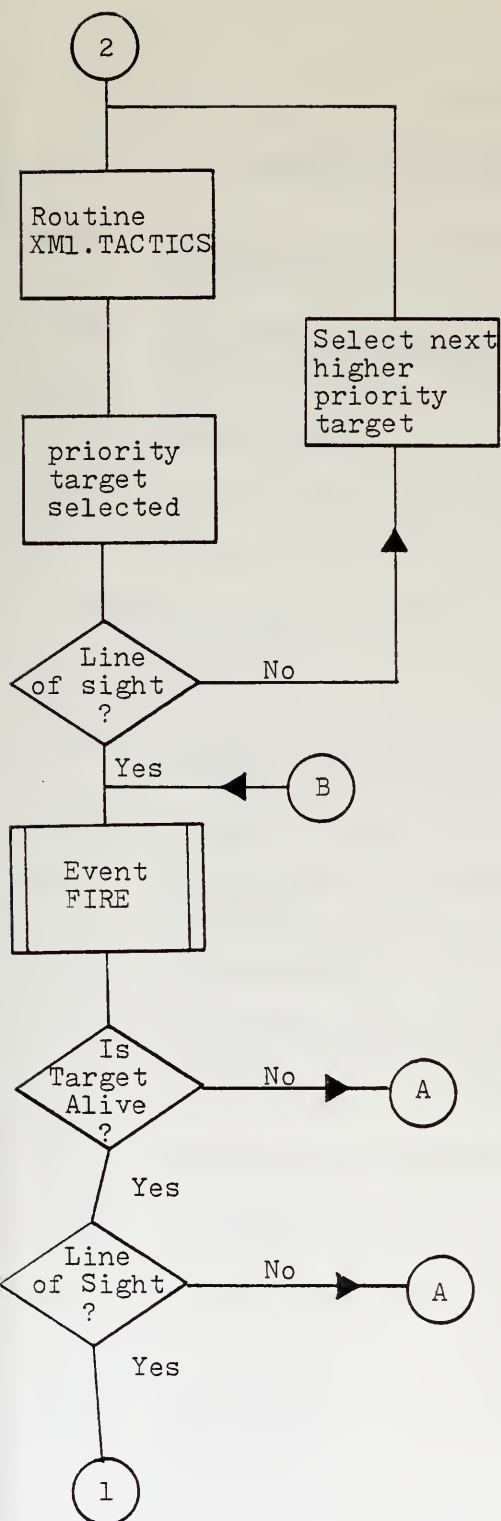
Event TARGET.SELECT cycles through the firer's list of currently detected targets. For the XM1, routine XM1.TACTICS is called to determine if any other platoon member is scheduled to fire at the XM1's highest priority target. The priority target is selected and line of sight is checked to see if it still exists between firer and target; if so, event FIRE is scheduled; if not, the next priority target is selected and the process is repeated. Event FIRE first checks to see if the selected target is still alive and line of sight still exists to the target; if so, the event IMPACT is scheduled to occur; if not, event TARGET. SELECT is scheduled.

239

APPENDIX B

GLOSSARY OF TERMS AND ABBREVIATIONS

APDS - Armor Piercing Discarding Sabot; an armor defeating
        tank round

ATGM - Anti-Tank Guide Missile; general term referring to
        wire guided missiles

BLUE - Term referring to friendly forces in battle or
        simulation

BMP   - Warsaw Pact mechanized infantry carrier

BN   - Abbreviation referring to Battalion organization

Bushmaster - 25mm automatic weapon mounted on U.S. Infantry
        Fighting Vehicle

CO   - Abbreviation referring to company organization

DIVAD - Division Air Defense Gun; a tracked air defense
        weapon soon to be introduced into U. S. inventory

DRAGON- A medium range, wire guided, anti-tank missile

FO   - Abbreviation referring to an artillery forward
        observer

HEAT - High Explosive Anti-Tank; a tank round

IFV   - Infantry Fighting Vehicle; new U. S. infantry
        carrier equipped with an ATGM system and Bushmaster

Inf   - Abbreviation referring to Infantry forces

M113 - Present generation infantry carrier

Mech - Mechanized; used normally to refer to mechanized
        infantry forces

MG   - Abbreviation used to refer to any machinegun

PLT   - Abbreviation which refers to a platoon organization

RED   - Term referring to enemy forces in battle or
        simulation

T72 - Warsaw Pact main battle tank

XM1 - New generation U. S. main battle tank

ZSU - Warsaw Pact air defense weapon system

BIBLIOGRAPHY

1. Defense Intelligence Agency, DDB-2680-40-78, <u>Handbook on the Soviet Armed Forces</u>, February 1978.

2. Department of the Army, FM 71-2, <u>The Tank and Mechanized Infantry Battalion Task Force</u>, June 1977.

3. Department of the Army, FM 100-5, <u>Operations</u>, July 1976.

4. Department of the Army, IAG-13-U-78, <u>Soviet Army Operations</u>, April 1978.

5. Johnson, G.D., <u>SIMSCRIPT II.5 Users Manual S/360-370 Version</u>, Consolidated Analysis Centers, Inc., 1974.

6. Kelleher, E.P., <u>Simulation of the Tactical Employment of Field Artillery</u>, M.S. Thesis, Naval Postgraduate School, Monterey, December 1977.

7. Kiviat, P.J., Villanueva, R., and Markowitz, H.M., <u>SIMSCRIPT II.5 Programming Language</u>, 2nd Edition, Consolidated Analysis Centers, Inc., 1973.

8. Needels, C.J., <u>Parameterization of Terrain in Army Combat Models</u>, M.S. Thesis, Naval Postgraduate School, Monterey, March 1976.

9. <u>SIMSCRIPT II.5 Reference Handbook</u>, Consolidated Analysis Centers, Inc., March 1976.

10. United States Army Combat Developments Command, Armor Agency, Report AR 69-2A(U), <u>The Tank Weapon System</u>, October, 1969.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Documentation Center
   Cameron Station
   Alexandria, Virginia 22314   2

2. Library, Code 0142   2
   Naval Postgraduate School
   Monterey, California 93940

3. Department Chairman, Code 55   1
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93940

4. Professor S. A. Parry, Code 55PY   5
   (thesis advisor)
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93940

5. LTC Edward P. Kelleher, Code 55Ka   1
   (thesis advisor)
   Department of Operations Research
   Monterey, California 93940

6. CPT E. G. Hagewood   1
   373 E Bergin Drive
   Monterey, California 93940

7. CPT W. S. Wallace   1
   373 C Bergin Drive
   Monterey, California 93940

8. Office of the Commanding General   1
   U.S. Army TRADOC
   Attn: General Donn A. Starry
   Ft. Monroe, Virginia 23651

9. Headquarters   1
   U.S. Army Training & Doctrine Command
   Attn: ATCG-T (Col. Ed Scribner
   Ft. Monroe, Virginia 23651

10. Headquarters   1
    U. S. Army Training & Doctrine Command
    Attn: Director, Analysis Directorage -
    Combat Developments (Col. Tony Pokorney)
    Ft. Monroe, Virginia 23651

11. Headquarters   1
    U. S. Army Training & Doctrine Command
    Attn: Director, Maneuver Directorate -
    Combat Developments (Col. Fred Franks)
    Ft. Monroe, Virginia 23651

12. Lt. General J. R. Thurman   1
    Commanding General
    U.S. Army Combined Arms Center
    Ft. Leavenworth, Kansas 66027

13. Director   1
    Combined Arms Combat Development Activity
    Attn: Col. Reed
    Ft. Leavenworth, Kansas 66027

14. Dr. Wilbur Payne, Director   1
    U.S. Army TRADOC Systems Analysis Activity
    White Sands Missile Range, New Mexico 88002

15. Director   1
    Armoed Combat Vehicle Technology Program
    Attn: Col. Fitzmorris
    U. S. Army Armor Center
    Ft. Knox, Kentucky 40121

16. Director   1
    Studies Division - Combat Developments
    Attn: Col. Burnett
    U.S. Army Armor Center
    Ft. Knox, Kentucky 40121

17. Col. Frank Day   1
    TRADOC Systems Manager - XM1
    U.S. Army Armor Center
    Ft. Knox, Kentucky 40121

18. MG Dick Lawrence   1
    Tank Forces Management Office
    Room 1A-871, The Pentagon
    Washington, D.C. 20310

19. Mr. David Hardison   1
    Deputy Undersecretary of the Army
    (Operations Research)
    Department of the Army, The Pentagon
    Washington, D.C. 20310

20. Director   1
    U.S. Army Material Systems Analysis Activity
    Attn: Mr. Will Brooks
    Aberdeen Proving Grounds, Maryland 21005

21. Command and General Staff College   1
    Attn: Education Advisor
    Room 123, Bell Hall
    Ft. Leavenworth, Kansas 66027

22. Headquarters, Department of the Army   1
    Office of the Deputy Chief of Staff
    for Operations and Plans
    Attn: DAMO-2D
    Washington, D.C. 20310